Using YOLO in Detecting and Localising Morepork Sounds

Supervisor

Monjur Ahmed
Arthur Do Valle

Submitted by

Name: Yanan Wang **ID:** 19489103

IMPORTANT
Submission of work that is not your own is treated as academic misconduct and may result in exclusion from the Waikato Institute of Technology. Penalties are identified in the Institutes Academic Regulations (a copy is available at the library or online).

I certify that this is all my own work, except for those parts identified for which references have been made.

Student Signature: Yanan Wang

Abstract

The research aims to implement and evaluate an advanced Convolutional Neural Network architecture, namely "You Only Look Once", to provide highly accurate data for the Morepork preservation projects. Implementing this technology is a complex task whose performance is affected by many factors. The research also seeks what the factors are and how they influence the results.

This research initiates from exploring the Convolutional Neural Network's definition and its procedures for detecting bird sounds. After exploration, the Design Science Research methodology is determined to guide the research conduction. A Design Science Research frame is adapted with five stages: 1) Identify the problems and limit the research scope; 2) Conduct a systematic literature review and formulate a design for the problems; 3) Implement the artefacts according to the design; 4) Evaluate the artefacts' results; 5) Generate knowledge from the process.

The literature review identifies three possible factors influencing the architecture's performance: presentation type, colourmap type, and CNN architecture. Accordingly, the author defines three research question with hypotheses to discuss the factors' influences. A full experimental design is conducted to verify the hypotheses. The research outcomes contain artefacts that contribute to the Morepork preservation projects and knowledge that directs future relevant research.

Keywords: AI, CNN, colourmap, detection, image processing, Morepork, sound, spectrogram, YOLO

Contents

Figures

Tables

Acknowledgements

The author would like to thank Dr Tim Hunt and the Cacophony project for their indispensable contribution to this work. Training a convolutional neural network in this research requires two essential while expensive and time-consuming tasks: recording a large number of audios and labelling them. Dr Hunt provided both. He allocated the author access to recordings in the Cacophony database. Also, he generously shared the author 18094 tags, for which he spent days listening to the audios. The author also wants to thank two supervisors Dr Arthur and Dr Monjur, for their instruction and help during the research.

1. Introduction

This chapter introduces the background, aims, and significance of this research. The bird preservation situation in New Zealand is firstly presented, where one problem emerges: the current solution in bird preservation projects performs poorly. The author examines different architectures of Convolutional Neural Networks (CNN) and chooses You Only Look Once (YOLO) as the solution to address the problem. This research is established with aiming to implement and assess the YOLO technique for one New Zealand native bird, Morepork. The outcomes of the research contribute to the development of both society and science. Finally, the last section illustrates this thesis' whole structure.

1.1. Background

New Zealand has been the home of more than 467 bird species. However, among over 200 native bird species, 68% face the threat of extinction (Forest & Bird, 2018). Monitoring the diversity and migration of bird species is an essential part of the bird conservation process (Mohanty, Mallik, & Solanki, 2020). Furthermore, birds live in a broad environment, making monitoring bird species beneficial for almost all conservation efforts (Kahl, Wilhelm-Stein, Klinck, Kowerko, & Eibl, 2018).

In forests and jungles, the light spread is easily blocked by leaves and branches. In contrast, sounds have a longer distance propagation without being occluded by objects in the middle (Koh et al., 2019), which makes audio detection a better choice than optical methods. Furthermore, scientists can use birds' sounds to diagnose their infested diseases such as respiratory infections or identify the change in their population size, retrieving first-hand information about climate change (Mohanty et al., 2020).

New Zealand is in the top place in bird recovery programmes globally (New Zealand Tourism, 2018), and monitoring bird sounds have been used in the nation's bird protection projects. The Cacophony Project aims to protect New Zealand native birds from the threat of introduced

predators, such as rats, stoats, and possums (The Cacophony Project, N.A.). The project uses a set of technologies to observe, identify and eliminate invasive predators, whose impact is evaluated by monitoring the bird songs density over time. In 2019, The project obtained the "Best Hi-Tech Solution for the Public Good" award.

1.2. Problem Statement

One of the Cacophony Project's monitoring targets is a New Zealand native bird named Morepork or "ruru". Morepork is a small, dark-brown owl widely distributed in the New Zealand forest (Morgan & Styche, 2012). Most of Morepork sounds can be cleanly identified as two short spurt syllables, the onomatopoeic of "more-pork" or "quork-quork" (New Zealand Birds Online, 2013). The majority of morepork calls occur at dusk and dawn due to their activity pattern, while very few calls are heard during the daytime (Brighten, 2015). The author chose morepork sounds as the research target because of their regular active period and distinguishable syllable patterns.

The Cacophony Project deploys mobile devices powered by a solar panel in the forest to capture morepork sounds in their active period. The captured recordings are uploaded to the Cacophony server. Technicians in the project use the Python programming language and machine learning such as Tensorflow and Keras to analyse the audios (Hunt, Nikora, & Blackbourn, 2019). In the analysing process, the researchers use a segmentation algorithm using signal strength variation to identify Morepork sounds (Hunt et al., 2019). As a result, the approach obtains a false-negative rate of 46% in the morepork detection task, which means it fails to recognise 46% of morepork calls in the given samples. Dr Tim Hunt mentions this figure as "disappointingly high".

1.3. Solutions

This research aims to tackle the stated problem and provide more accurate information for the projects with the advanced CNN technique. A CNN is a class of deep neural networks that take in images and analyses their salient features by learnable weights and biases (O'Shea & Nash, 2015). It has been showing remarkable performance in identifying bird species by inputting

birds' acoustic representations. Bird Cross-Language Evaluation Forum (BirdCLEF) is a competition where worldwide participants classify birds' species according to their calls. Sprengel, Jaggi, Kilcher, and Hofmann (2016) won the competition in 2016 by designing a CNN model. Since the first winning in 2016, more advanced CNN-based architectures have achieved the best performance in the competitions (Kahl et al., 2020).

CNN has various architectures to detect features in images. As a new CNN object detection approach, YOLO predicts class probabilities and localises features by bounding boxes from looking and evaluating full images only once (Redmon, Divvala, Girshick, & Farhadi, 2016). The YOLO approach has been shown efficient performance in detecting wild bird tasks. For example, Hong, Han, Kim, Lee, and Kim (2019) detected wild birds by using different CNN models, including Retinanet, Region-based Fully Convolutional Network, Faster Region-based Convolutional Neural Network (R-CNN), Single Shot MultiBox Detector (SSD), and YOLO. In their comparison, YOLO was the fastest among all models. Zhang, Yang, Tang, and Liu (2018) compared the performance and speed among Fast R-CNN, Faster R-CNN, SSD, Context & R-CNN, and YOLO. As a result, YOLO achieved the highest accuracy while used the shortest time.

1.4. Research Aims

Implementing a neural network requires setting a wide range of hyperparameters. Neural networks are sensitive to the hyperparameters settings (Montavon, Orr, & Müller, 2012), and the optimisation of the settings decisively determines the neural network's speed and performance (Murugan, 2017). Thus, a proper setting of the parameters yields better performance and also speeds up the training process. However, optimal configurations are hard to determine, and the tunning process is computationally and timely expensive (Domhan, Springenberg, & Hutter, 2015). Therefore, the author launched this study to utilise and assess YOLO in detecting Morepork sounds. The research has two main aims with sub-targets:

- Creating an artefact to solve the research problem
    - Implement a project of using YOLO models to detect morepork sounds

    - Compare different combinations of the factors' candidate values to find the optimal method
- Generating knowledge from the artefact implementation process
    - Find out the determining factors in the hyperparameters of the project
    - Determine each factor's influence on the project's results

1.5. Research Significance

The research outcomes are artefacts to tackle the real-world problems and knowledge generated from the artefact's implementation process. The research contributes to both bird preservation projects and future related research.

The bird preservation projects require accurate information to evaluate their past effort and make future decisions. However, the existing approach to provide information in the Cacophony project has poor performance. In comparison, the artefacts created in the research provide much higher accurate information for the Cacophony project than current methods. Besides, it is easy to utilise the promising technology for other New Zealand native bird species by following the same implementation process.

The report comprehensively explains bird sound processing and the state-of-art YOLO technology, helping further study in this area. The author also shares the results of the experiments and reveals three determine factors' influence in the YOLO architecture. When designing a new model, the designers can take these findings into consideration. Also, other researchers can compare their models' results with this one to evaluate their models' performance.

1.6. Report Structure

Figure 1 illustrates this report's structure and logical connections among chapters. The first chapter describes the problems in the given background and then briefly introduces the CNN and

YOLO technologies to address the issues. Next, the research aims, significance and contributions are presented.

Chapter 2 discusses a systematic literature review to explore the process and factors in the CNN solution. The chapter firstly discusses the review questions, criteria, and search keywords. The review reveals a general procedure for implementing the CNN techniques in the bird detection tasks. In the process, several factors are found to influence the results. As this research focuses on YOLO architecture, the architecture's concepts and work theory are especially illustrated. Finally, based on the review findings, the research questions are generated to search for the factors' influence.

Chapter 3 presents the description and justification of the adopted research methodology, based on which a research process is designed. The chapter then lists hypotheses for the three research questions. The relationship of the variables, research questions, hypotheses, and methodology is shown in a framework at the chapter's end.

Chapter 4 elaborates on the experiment used to collect data for the research. The experiment's structure and working flow are illustrated in the first section. The whole working flow contains three main stages. The first stage, data processing, contains four steps, which are described in the second section. The third section discusses the second stage of the experiment, focusing on telling the environment, technical terms, and variable settings. Finally, the last section discusses the evaluation terms and methods.

The results of the experiment are shown in raw type. In Chapter 5, the data is converted into readable information, which is further visualised in charts and graphs. By the patterns in the figures, all factors' influence is detailed compared. From the comparison results, the author validates all hypotheses and gives recommendations in Chapter 6. Finally, Chapter 7 generates conclusions based on the research findings, including the research's achievements and limitations. In addition, the chapter lists directions for further study.

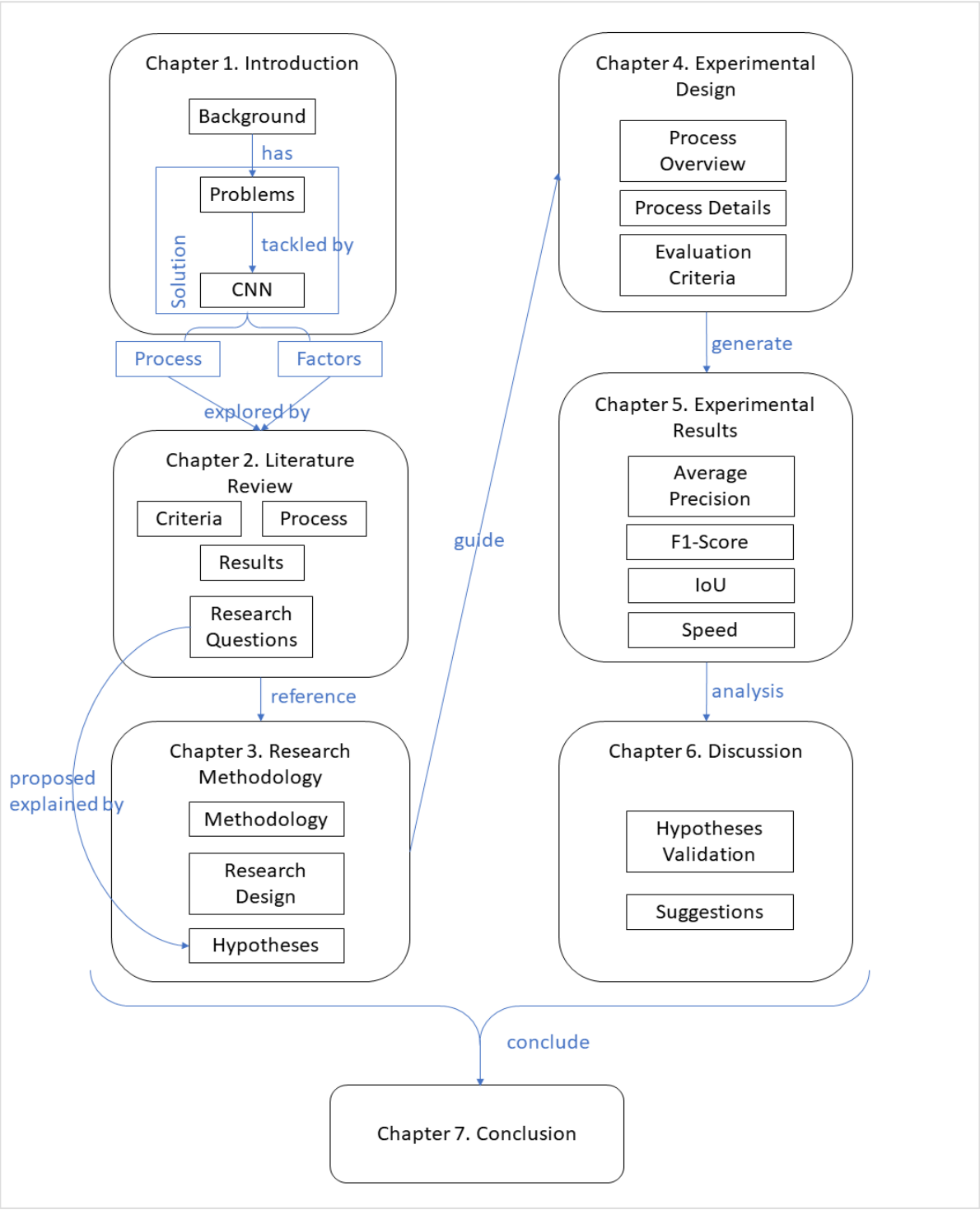


Figure 1. Structure of the Report

1.7. Conclusion

Monitoring birds sound is a vital task for New Zealand bird preservation projects. CNN provides a much more promising performance in detecting bird sounds than existing approaches. YOLO is an outstanding architecture among all CNN models for object detection tasks. The author aims to implement and analyse this advanced technology. A comprehensive understanding of the YOLO solution is required before the implementation. The next chapter presents a systematic literature review related to YOLO and bird sound detection tasks.

2. Literature Review

This chapter presents the findings from the relevant literature review. The review firstly explores the concepts of CNN and YOLO and then reveals how these techniques process bird sound detection tasks. In the process, three factors are identified with a potential effect on the results. However, whether these three factors influence this research's scenario is unknown, and therefore research questions regarding the factors are formulated.

2.1. Review Process

As shown in Chapter 1.4, the research aims to implement YOLO in detecting Morepork sounds and analyse influence factors in the implementation. As YOLO was a new CNN architecture when the literature review started, there were limited articles relative to YOLO and bird sounds detection. Besides, many researchers compared YOLO with other CNN architectures in their research, with a paper title contains CNN. Therefore, the author expanded the search scope to all CNN models.

The review process contains three steps: searching, filtering, and analysing. In the first step, search keywords (Table 1) are generated to search for journal articles based on the review questions. Then, the author forms exclusion criteria (Table 2) to disqualify studies from the searching result. Finally, after filtering founded essays by exclusion criteria, the researcher processes all the selected articles, extracts relevant content from the articles, and analyses essential information from the content.

Table 1. Search Keywords

No.	Content
1	CNN or ConvNet or Convolutional Neural Networks
2	classify or classifying or classification or detect or detecting or detection or recognise or recognising or recognition or pattern

No.	Content
3	sound or acoustic or audio or call or song
4	bird

Table 2. Exclusion Criteria

No.	Category	Description
1	Publish time	Studies that have been published for more than five years
2	Language	Studies that are not reported in the English language
3	Genre	Studies that are not published in journal articles or conference papers
4	Peer review	Journals that have not been peer-reviewed
5	Duplication	Duplicate reports for the same study

Overall, the literature review map composites 61 searched articles, including 23 implementations for birds, 6 practices on other objects, 10 competitions papers, 14 original masterpieces of the CNN milestone concept, 3 papers about YOLO, 1 article about spectrograms, and 5 other articles related to the research (Figure 2).

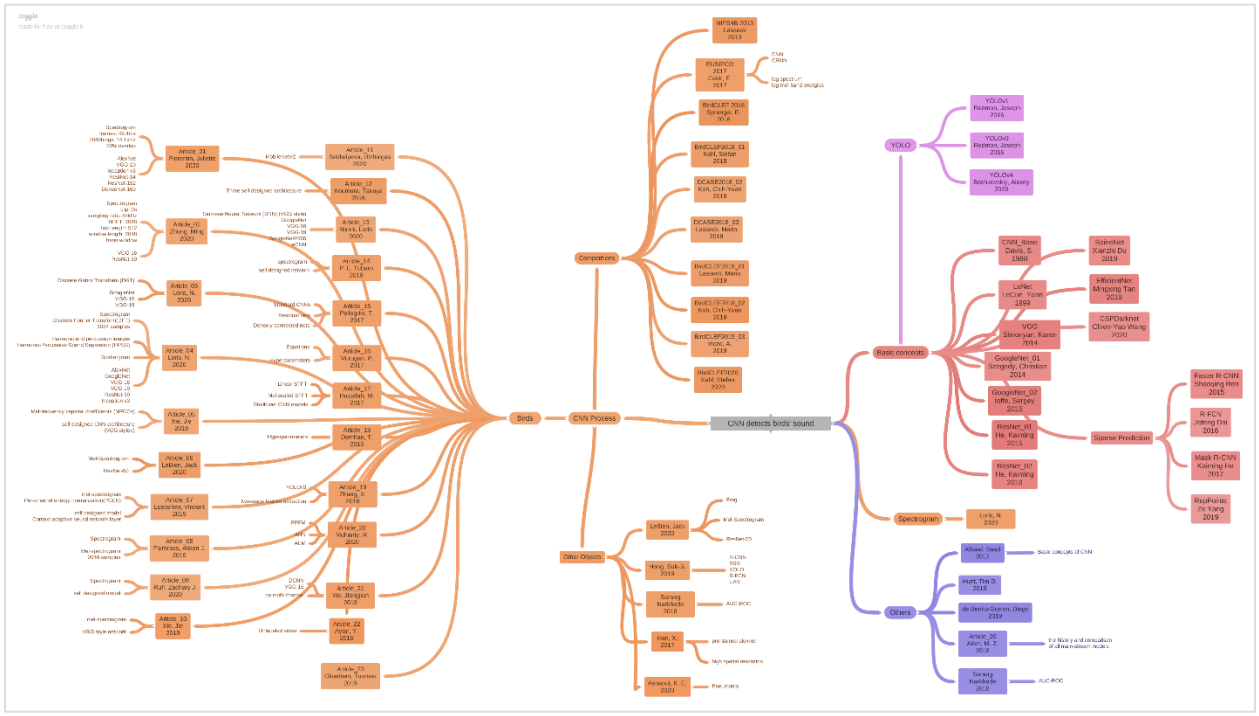


Figure 2. Literature Review Map

2.2. CNN Concepts

This section presents the conception of CNN and YOLO, including their definitions, architectures, and evolving history.

2.2.1. CNN Architecture

As an alternative type of neural networks, CNN can reduce spectral variations and recognise special and temporal correlations in signals (Sainath, Mohamed, Kingsbury, & Ramabhadran, 2013). Alom et al. (2018) illustrate the overview of CNN architecture consisting of two parts – features extraction and classification, as shown in Figure 3. The structure begins with an input layer, where pre-defined-size images are taken as train data or evaluation data. Next, the multiple convolutional layers and pooling layers extract the input images' features. A CNN architecture ends by the output layer. Fully connected layers firstly flatten the generated feature maps, and

then the output layer makes predictions from the flattened data. Alom et al. (2018) categories the layers in the architecture into three types according to their functions:

1. Convolutional Layer

   The convolutional layer extracts features from input images by multiplying the input data matrix with a set of learnable weights. The output of this layer is called feature maps.
2. Sub-sampling Layer

   The sub-sampling layer, commonly known as the pooling layer, executes the downsample operation to shrink the feature map size. There are two main operation types: max-pooling and average pooling.
3. Classification Layer

   The classification layer makes predictions from the extracted feature maps.

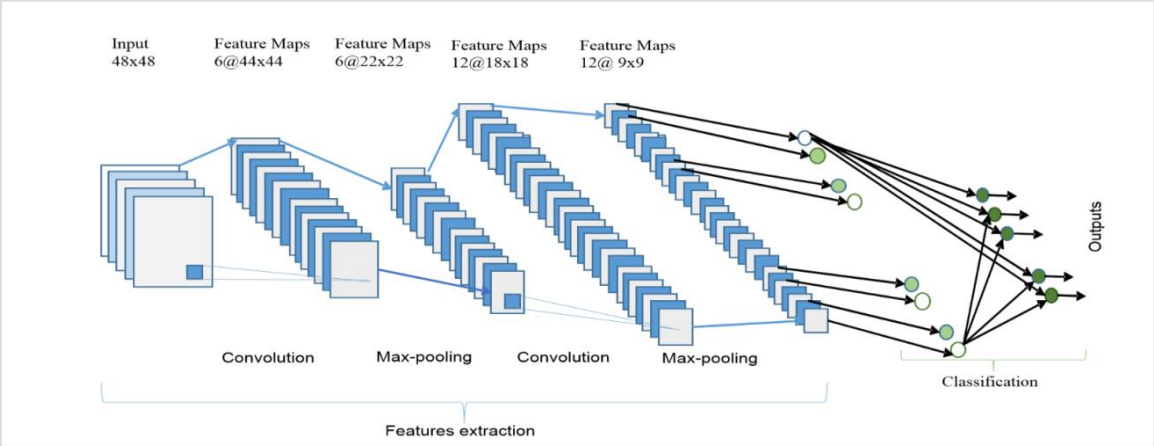


Figure 3. The Architecture of CNN (Alom et al., 2018)

2.2.2. CNN History

Figure 4 reveals the CNN developing history. Since first invented by Fukushima (1988) to recognise handwriting, the CNN models have never stopped evolving. Krizhevsky, Sutskever, and Hinton (2012) introduced the first deep CNN, named AlexNet. Based on AlexNet, the Visual Geometry Group (VGG) from Oxford invented the VGG architecture, making the architecture

"deeper". The VGG architecture won the competition of ImageNet LSVRC-2014 (Simonyan & Zisserman, 2014). Based on the 19-layer VGG architecture, He, Zhang, Ren, and Sun (2016) invented ResNet, a very deep network using residual connections.

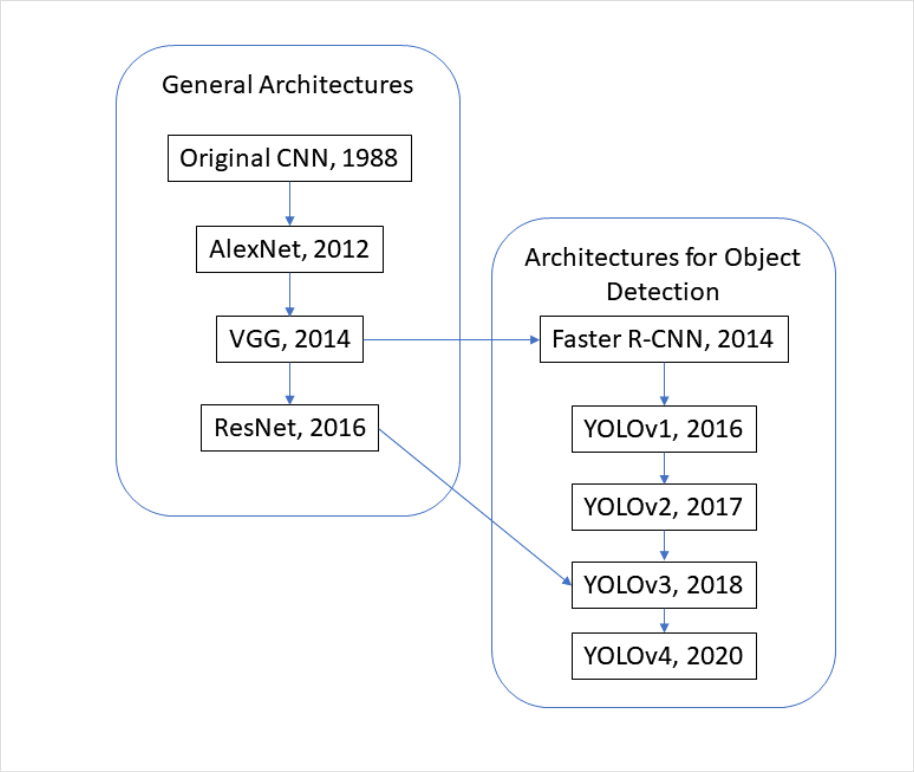


Figure 4. CNN Involving History

Using VGG architecture as the backbone, the Fast Region-based Convolutional Network method (Fast R-CNN) was introduced, especially for the object detection tasks (Girshick, 2015). Fast R-CNN cannot make predictions strictly from the image; Instead, it requires repurposes classifiers to perform before classification. In 2016, Redmon et al. (2016) introduced the first version of YOLO to remove the repurposes classifiers step. In the following two years, Redmond released the 2nd (Redmon & Farhadi, 2017) and 3rd (Redmon & Farhadi, 2018) versions, with incremental improvement on each new version. The 3rd version (YOLOv3) adopts the residual connections

from the ResNet in its architecture. In 2020, Bochkovskiy, Wang, and Liao (2020) released the 4th version (YOLOv4) and claimed it outperforms YOLOv3 in both accuracy and speed.

2.2.3. One CNN Architecture for Object Detection: YOLO

As a new CNN object detection approach, YOLO predicts class probabilities and localises features by bounding boxes simple from one evaluation of looking full images (Redmon et al., 2016). Other detection methods, such as DPM and R-CNN, must decide domains from images first and then predict the domains. In comparison, YOLO can recognise and localise features by evaluating (looking) full natural images only once. Besides, YOLO can be optimised end-to-end directly on the detection tasks as its pipeline is a signal network.

Figure 5 illustrates the three steps in the YOLO working procedure (Redmon et al., 2016). It firstly resizes images to a specific size **N** x **N** (**N** is a multiple of 32) and then divides the input into **S** x **S** grids. Next, for each grid, YOLO makes three predictions by three bounding boxes. Each box contains the following predicted information: the probability of each class, the confidence score of that class, and the object's positions. Finally, YOLO filters the predictions for the same object and choose the one with the highest confidence.

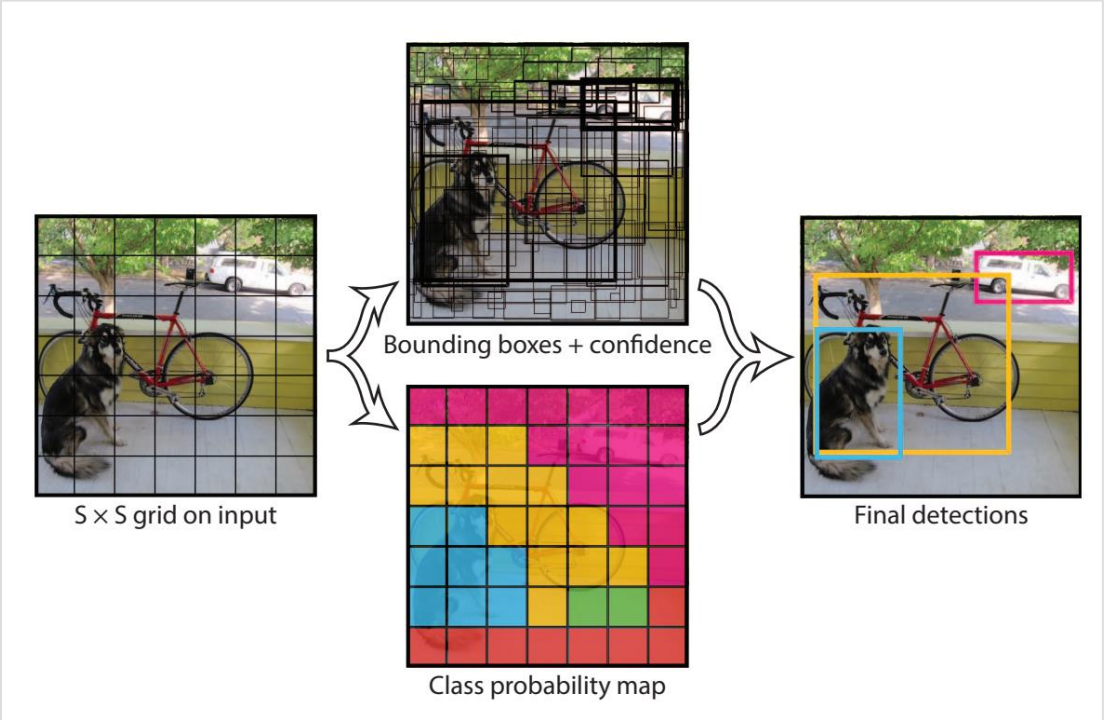


Figure 5. The YOLO Working Process (Redmon et al., 2016)

2.3. The Process of Using CNN in Sound Classification Tasks

Implementing CNN contains three main steps 1) processing audios into visual presentations, 2) using presentations to train CNN models, and 3) evaluating the trained models. The three steps are detailed displayed as follows:

Step 1. Pre-processing audios

This pre-processing process starts with downloading video recordings and labelling their tags. Then the videos are processed into image representations, mainly spectrograms and their derivatives (Incze, Jancsó, Szilágyi, Farkas, & Sulyok, 2018; Loris et al., 2020; Nanni, Rigo, Lumini, & Brahnam, 2020; Xie, Ding, Li, & Cai, 2018). Next, these researchers divide processed visual presentations into two groups, the training group to train CNN models and the testing group to evaluate the performance of trained models.

Step *2. Training CNN*

Literature shows that programmers develop either an existing architecture or a self-designed CNN model to "learn" patterns in the training group data. Among all CNN models, VGG, Inception Networks and ResNet are the most popular existing architectures (Nanni, Maguolo, & Paci, 2020; Zhong et al., 2020). Some researchers like Nanni (2020), Tubaro (2019) and Xie (2019) designed their models using the concepts of multiple popular models.

LeCun, Haffner, Bottou, and Bengio (1999) introduced the gradience in the CNN training epochs. One training epoch contains the following processes. Firstly, the CNN models take processed images as input from the input layers. A CNN architecture has parameters that process the input images to specific numbers as results. Next, the calculated results are compared with the manually labelled tags to get the difference, called Loss. Finally, based on the Loss, the CNN uses the gradient calculation to update its parameters to reduce the Loss.

A CNN model requires numerous epochs to reduce its Loss to a tolerable level. The epochs number varies based on the architecture and detected objects. Valentyn Sichkar (2021) recommends the YOLO training epochs as 2000 times as the total detected objects' classes.

Step *3. Evaluating results*

The trained CNN processes video presentations into a list of numerical marks, which show no direct indication of its performance. Different approaches were used to evaluate the results. He et al. (2016), Fairbrass et al. (2019), and Redmon et al. (2016) used the average precision to evaluate models. Accuracy, precision and recall were shown in the research of Florentin, Dutoit, and Verlinden (2020), Ruff et al. (2020). Sensitivity, specificity and the area under a curve were also used to denotes the model performance by Zhong et al. (2020), Nanni, Rigo, et al. (2020). The term F1 Score was utilised by Xie and Zhu (2019) and Oikarinen et al. (2019). In the competitions with more than 1000 classes, Simonyan and Zisserman (2014), Szegedy et al. (2014) and Ioffe and Szegedy (2015) adopted the Top-1 error and Top-5 error to compare different model's performance.

2.4. The Influential Factors

In practice, researchers mainly put effort into tuning various hyperparameters to optimistic their models' performance. Three factors are highlighted among the hyperparameters.

2.4.1. Factor 1: Presentation Type

Literature review shows that implementations with various representation types obtain a notable distinction in their results (Xie et al., 2018). Audios can be presented in multiple ways for the CNN training. This section lists three types of audio representations, namely raw waveform, linear spectrogram and Mel spectrogram. Figure 6 shows all three types extracted from one same voice recording.

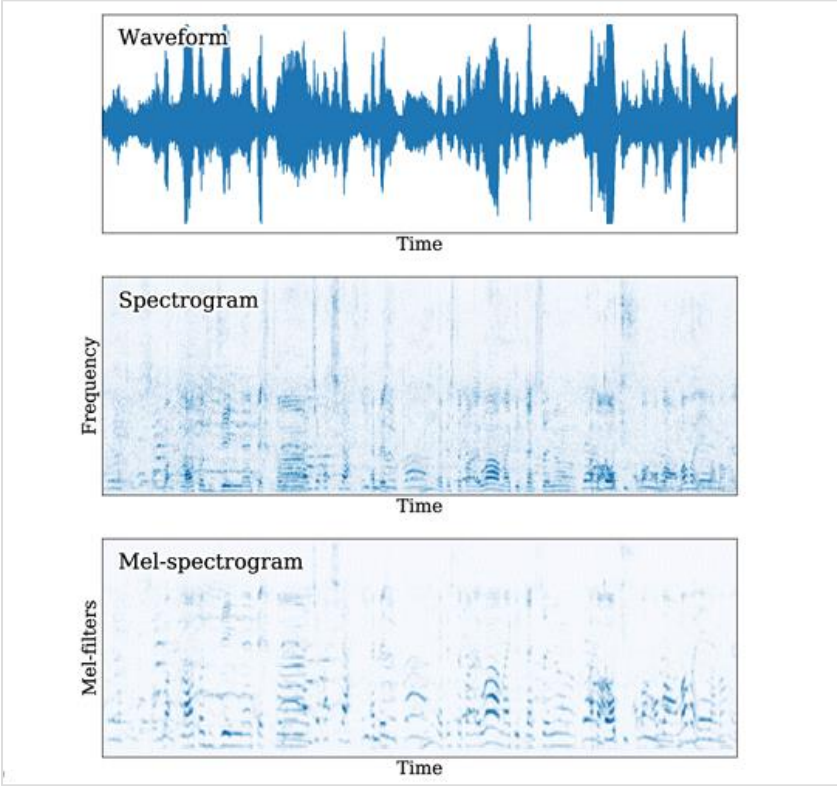


Figure 6. A Voice Segment Presents in the Waveform, Linear Spectrogram and Mel-Spectrogram (de Benito-Gorron, Lozano-Diez, Toledano, & Gonzalez-Rodriguez, 2019)

Type 1. Raw Waveform

Some researchers adopted the raw waveform for their CNN models (Fu, Tsao, Lu, & Kawai, 2017; Golik, Tüske, Schlüter, & Ney, 2015; Oord et al., 2016; Palaz, Collobert, & Doss, 2013). The raw waveform is sampled directly from audios, and it presents signals in the time domain (Fu et al., 2017). This type presents all digits in the audios losslessly.

Some models use this type to analyse sounds. For example, WaveNet (Oord et al., 2016) and SoundNet (Aytar, Vondrick, & Torralba, 2016) take this type of presentation as input. The WaveNet is used to predict the most likely subsequent samples in human speech and music sequences. In addition, SoundNet can classify the scenes or identify the objects in videos by their soundtrack.

The two models do not require complex transformation before operating the audios, providing flexible audio analysis methods. However, their performance even disappoints their inventors. For example, WaveNet's process is slow and expensive (Oord et al., 2016). While on the other hand, the SoundNet obtained a lower accuracy of 32.4% by sounds than by images (49.4%) in classification tasks (Aytar et al., 2016). In short, the two methods are either ineffective or inefficient.

Type 2. Spectrograms

Spectrograms represent the signal's spectrum of frequencies varying over time (Sejdić, Djurović, & Jiang, 2009). Figure 7 shows three dimensions in a spectrogram image: the X-axis (width) denotes the time domain; the Y-axis (height) presents the frequency; and the third dimension, colour, means the signal's energy or amplitude. The warm colours such as red and yellow designate strong amplitudes, while the cool colours like cyan and blue refer to weak amplitudes.

The Mel scale is a particular unit proposed by Stevens, Volkmann, and Newman (1937) to rescale sound pitches according to human's perceived magnitude. Human ears are more sensitive more to low frequency than the high frequency in the voice. Accordingly, the conversion

stretches the low-frequency part and squeeze the high-frequency section. By presenting amplitude from the linear scale to the Mel scale, linear spectrograms are converted to Mel spectrograms, showing more details in the low-frequency area.

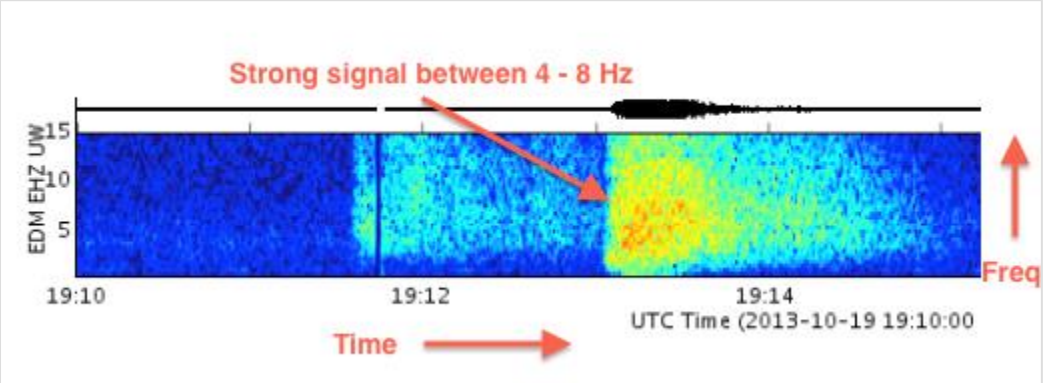


Figure 7. A Spectrogram Contains Three-Dimension Information (Pacific Northwest Seismic Network, N.A.)

Incze et al. (2018) and Huzaifah (2017) designed experiments to compare the performance of linear and Mel spectrograms. In their results, the Mel spectrogram achieved better performance than the linear counterpart, with higher accuracy of 3.23% and 7.4% on average, respectively.

2.4.2. Factor 2: Colourmap Type

The spectrogram images can be presented in different colour scales. Greyscale is a one-channel representation where white is denoted as 0, black as 1, and the shades of grey in between. A greyscale image is shown in a linear white-black colour (Figure 8). The Jet colourmap is a three-channel representation that denotes each colour pixel by three numbers. The three channels, namely red, green, and blue, show images on a colourful scale (Figure 9).

Figure 8. The Colour Range of Greyscale

Figure 9. The Colour Range of Jet

Incze (2018) compared different colour maps of representation images and found the RGB spectrograms overperform the greyscale counterparts. His team pre-processed sound into the grayscale colour map and jet colour map (Figure 10). As a result, they achieved better accuracies and recalls from the RGB images than the one-channel representations in all 2-class, 10-class and 50-class experiments (Figure 11).

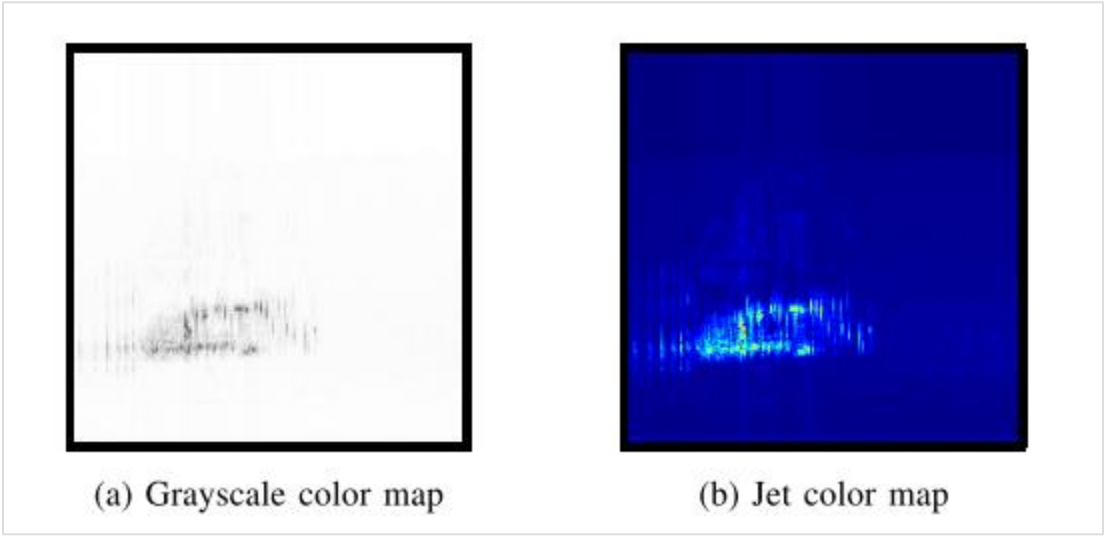


Figure 10. Spectrograms of One Bird Sound Clip, Rendered in Greyscale (a) and Jet (b) Colourmaps (Incze et al., 2018)

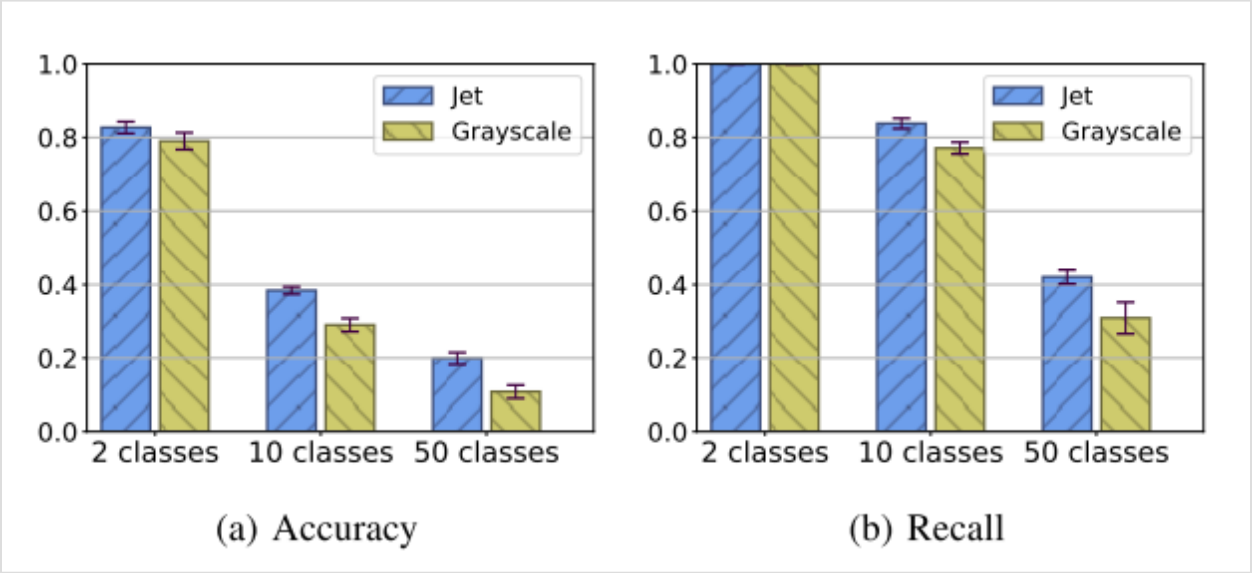


Figure 11. Results Comparison by Accuracy (a) and Recall (b) Between Two Different Colour Maps (Incze et al., 2018)

2.4.3. Factor 3: CNN Architecture

CNN has been widely used in bird sound recognition, detection, and classification tasks, with different architecture achieving various performances. The architectures are composed of many elements, such as the learning approaches, the depth (He et al., 2016), and the size (Krizhevsky et al., 2012). Different models have various settings in the learning methods and designs in size and depth.

As described in the previous chapter 2.2.2, the YOLO family models were invented on their predecessor architectures and took new concepts in their architecture. Bochkovskiy et al. (2020) compared the speed and accuracy of different state of the art object detector models and obtained the best accuracy and speed using YOLOv4 (Figure 12).

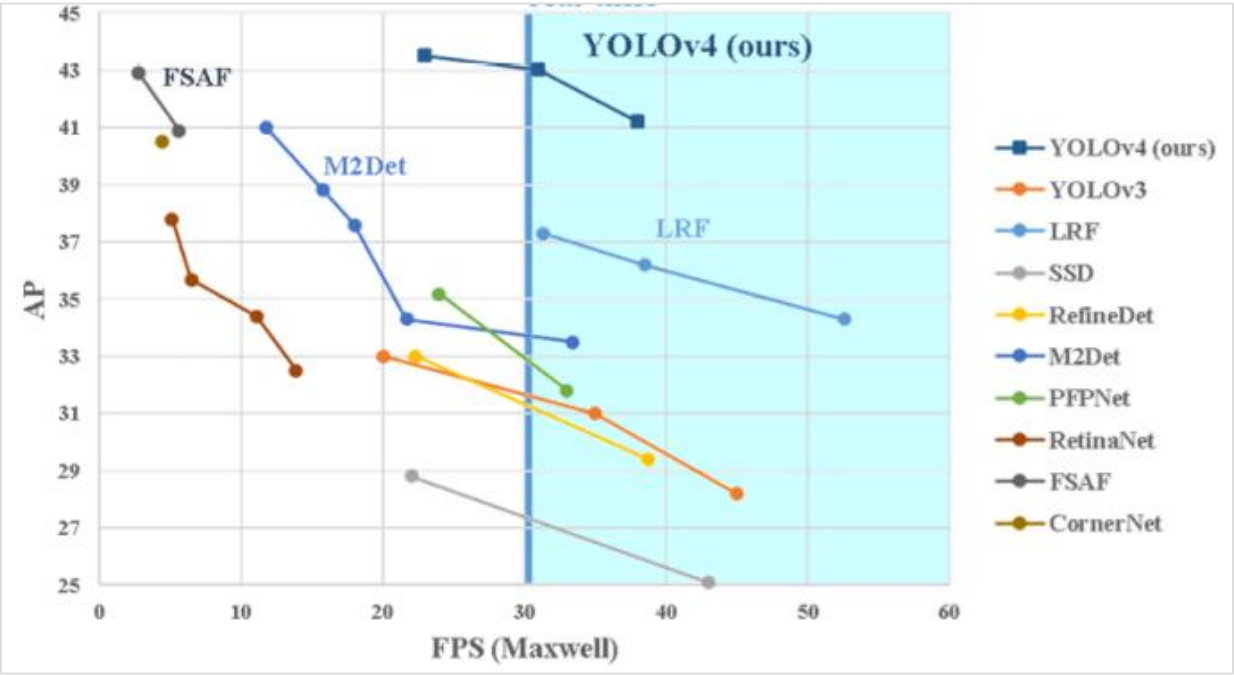


Figure 12. Comparison of Average Precision and Speed of Different Object Detector Models (Bochkovskiy et al., 2020)

2.5. Research Questions

The literature review reveals three factors in the general CNNs' implementation process. Whether the same factors influence the performance of using YOLO models in detecting Morepork calls is still unknown. Therefore, this research aims to explore the factors that influence the performance of utilising YOLO in detecting Morepork sounds. The three factors are named presentation type, colourmap type, and CNN architecture. Accordingly, the researcher formed three research questions (RQ) to reveal the relationship between these factors and the model's performance.

RQ 1: Does the presentation type influence the performance of using YOLO in detecting morepork sounds?
RQ 2: Does the colourmap type influence the performance of using YOLO in detecting morepork sounds?

RQ 3: Does the architecture influence the performance of using YOLO in detecting morepork sounds?

All research questions share the same background: using YOLO in detecting morepork sounds. The three questions highlight the three variables in this background. As discussed in section 2.2.1, CNN can recognise silent features from signals' images. Therefore, if given images with diverse representations or colourmaps, the YOLO model is anticipated to return different results. This assumption forms the RQ1 and RQ2. Various YOLO models have distinct layers and parameters, which means they process the images in divergent ways. However, from the literature review, the author found that every architecture has a ceiling performance. Therefore, there may exist a chance that the YOLO models can achieve ceiling precision without a noticeable difference. RQ3 is formed due to this consideration.

2.6. Conclusion

CNN has plenty of architectures to recognise patterns in images, and YOLO is one advanced approach specialised to detect and localise objects. By providing the image presentations of audios, the 4th version of YOLO can detect patterns in the presentations in real-time with reasonable accuracy.

When utilising a CNN model in acoustic tasks, many factors may affect the performance. The efforts in tuning these factors have made this technology more and more accurate in detecting bird sounds. However, CNN's implementation for one task distinguishes that from another task (Fang, Ma, Zhang, Zhang, & Bai, 2017). This research aims to find the relationship between the factors and the performance when implementing the technology for one New Zealand native bird, moreporks. The next chapter presents the research methodology for the research aim.

3. Research Methodology

This chapter presents the methodology of this study. The research procedure is designed based on the principles and a basic model of Design Science Research (DSR). The first section introduces the popularity of the methodology, lists its contributions, and discusses how it fits the research. In the previous chapter, three research questions are formed to address the research problems. For each research question, there assumed two hypotheses. The relationship of the questions and hypotheses are listed at the end.

3.1. DSR Methodology

DSR is an outcome-based research approach that provides specific guidelines to evaluate and iterate IT artefacts to solve particular problems (Hevner, March, Park, & Ram, 2004). DSR has a long tradition as the paradigm in information system (IS) research in many countries. It dominates the German-speaking countries and is popular in France, Italy and Netherlands (Winter, 2008). Indulska (2010) found a magnificent increase in adopting DSR in IS research after identifying the research methodology in 1037 published IS papers.

According to the research of Baskerville (2018), the DSR is an increasingly important research approach in IS as it makes two dominant contributions from a project: the creation of something (design artefacts) and the process of creation (design theories). Therefore, DSR research can address a practical problem and meet publications' academic requirements simultaneously. Baskerville (2018) lists key contributions of DSR as following:

1. DSR provides the design for a cutting-edge IT artefact.
2. DSR introduces the artefact to an application context with measurable improvements.
3. DSR extends and generalises the knowledge contribution of a project.

DSR is adopted as the research methodology because its listed characteristics perfectly address the research problems. Firstly, YOLO is a state-of-art technology whose implementation requires

highly on time and resources. Besides, the research involves developing an application to address real-world problems, through whose process essential findings are generated. The DSR paradigm gives essential problem-solving steps to solve application problems, evaluate solution designs, and grow knowledge from the application.

3.2. Research Process

As DSR is chosen for the research, Figure 13 presents the research process defined by SDR. Based on the original description by Vaishnavi and Kuechler (2015), the research is designed with five main stages, awareness of problems, suggestion, development, evaluation and conclusion. The process can be executed sequentially, but when new questions arise in the third and fourth stages, the execution moves back to the first stage.

Stage 1. Awareness of Problems

The research starts with limiting the research scope. Next, research problems in the scope are identified. The magnificence of tackling all the problems is evaluated. The problem with the most significant contribution is chosen as the research target. With clear research aims, the author conducts preliminary searching to find candidate solutions. After comparing all solutions by their costs and outcomes, the researcher chooses the most effective and feasible solution. As a result, the research proposal is formed to implement and evaluate the specific artefacts to address the research problem.

Stage 2. Suggestion

Vaishnavi and Kuechler (2015) discuss that DSR uses a series of analytical and synthetic approaches and perspectives for performing research in IS. Therefore, fundamental knowledge of techniques and perspectives are required before implementing the solution. This stage conducts a literature review to obtain a historical perspective of the solution and consult pioneers' work in the research field. The author gathers necessary data related to the solution, analyses information for the data, and generates knowledge based on the information from the literature review. The

generated knowledge leads to form a specific suggestion. More literature review is required to get comprehensive knowledge of the suggestion. Finally, a design theory is adopted, and a tentative experimental design is created based on a thorough understanding and pioneers' work related to the suggestion.

Stage 3. Development

This stage focuses on implementing the artefact. The implementation process follows the process of adopted design theory. In the process, the author uses technical information and scientific principles in the tentative experimental design. Suppose new problems out of the research boundary emerge in the developing procedures. In that case, the research goes to the first step and refine its proposal. The oriental aims may vary based on the practical issues but always stick to tackle the problems. The output of this step is an artefact that provides data for the research.

Stage 4. Evaluation

After development, artefact performance data becomes available. Evaluation criteria are first designed based on the technology used in the experiment. Then the artefact's performance is evaluated and visualised by criteria and relevant techniques. This step presents the data in different genres, like tables, charts, and figures. Like the previous step, some findings that change the view of the solution can trigger refining the research targets and aims. At the end of this stage, the author generates an efficient and feasible solution for the research problem.

Stage 5. Conclusion

Goldkuhl (2004) argues that DSR research aims for knowledge building and knowledge growth through development. Therefore, this stage focuses on generating knowledge from the experiment. The author describes the experiences and obstacles in the experimental implementing process, illustrates different models' performance comparison, discusses the reasons behind the comparison results, and previews future research.

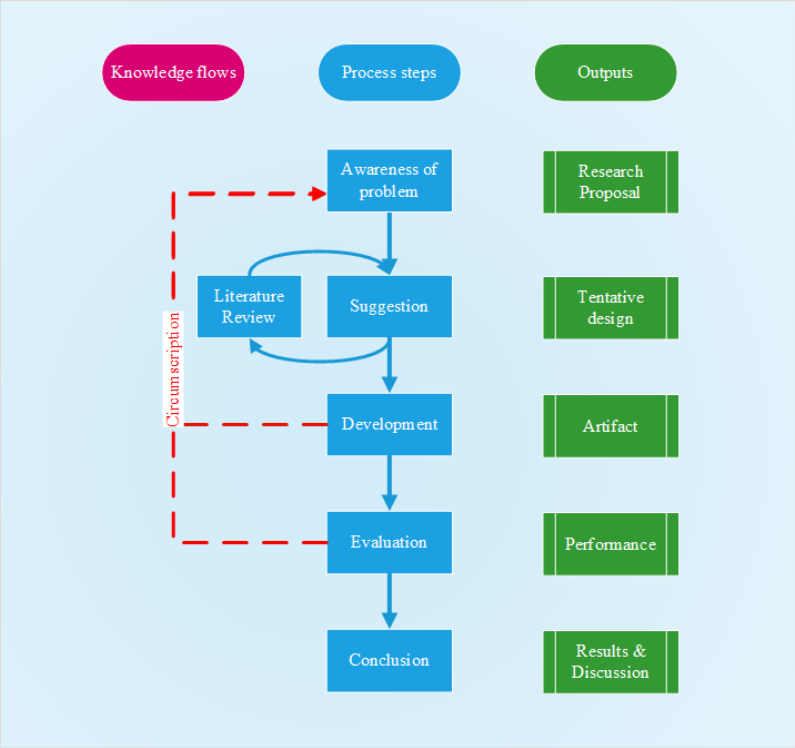


Figure 13. Research Methodology Model (Vaishnavi & Kuechler, 2015)

During the research process, the author revised the research proposal several times. After the research problem was identified, the research aimed to adopt general CNN architectures as the problem solution. Therefore, one search keyword was defined as "CNN" at the literature review stage, and the review investigated all CNN models. As a result, the author proposed the ResNet and GoogLeNet architectures to design the experiment.

In the developing stage, other architectures emerged that were specially designed for object detection. Two mainstream architectures were Single Shot MultiBox Detector (SSD) and YOLO. Considering the object detective architectures show more information than general architectures, the author changed the research proposal to "Using SSD and YOLO as solutions and compare their performance". Correspondingly, a second literature review related to these two architectures was launched.

The second review reveals that, compared with SSD, YOLO yields higher accuracy but requires more execution time (Liu et al., 2016). In addressing the research problems, accuracy is the primary consideration, while time cost can be ignored. Therefore, the author limited the research target to YOLO. The research aims were finally fixed in this step, and a third literature review was conducted, focusing on only YOLO architectures.

3.3. Research Questions, Hypotheses, and the Theoretical Framework

For each research question, there form two hypotheses (Table 3). The first positive hypothesis assumes the related factor affects the result, while the second directional hypothesis limits the experimental scope.

Table 3. Hypotheses for the Variables and Research Questions

Independent Variables	Research Question	Hypothesis
Factor 1: Presentation type	RQ 1	H1: The presentation type impacts the performance of using YOLO in detecting Morepork sounds. H2: Between two presentation types, linear spectrogram and Mel spectrogram, the Mel spectrogram makes YOLO achieves better performance in detecting Morepork sounds.
Factor 2: Colourmap type	RQ 2	H3: Colourmap type impacts the performance of using YOLO in detecting Morepork sounds. H4: Between two colourmap types, Greyscale and Jet, the Jet colourmap makes YOLO achieves better performance in detecting Morepork sounds.
Factor 3: Architecture	RQ 3	H5: Architecture impacts the performance of using YOLO in detecting Morepork sounds.

Independent Variables	Research Question	Hypothesis
		H6: Between two architectures, YOLOv4 and YOLOv4-tiny, the YOLOv4 architecture makes YOLO achieves better performance in detecting Morepork sounds.

The theoretical framework (Figure 14) shows the relationship among variables, research questions and hypotheses.

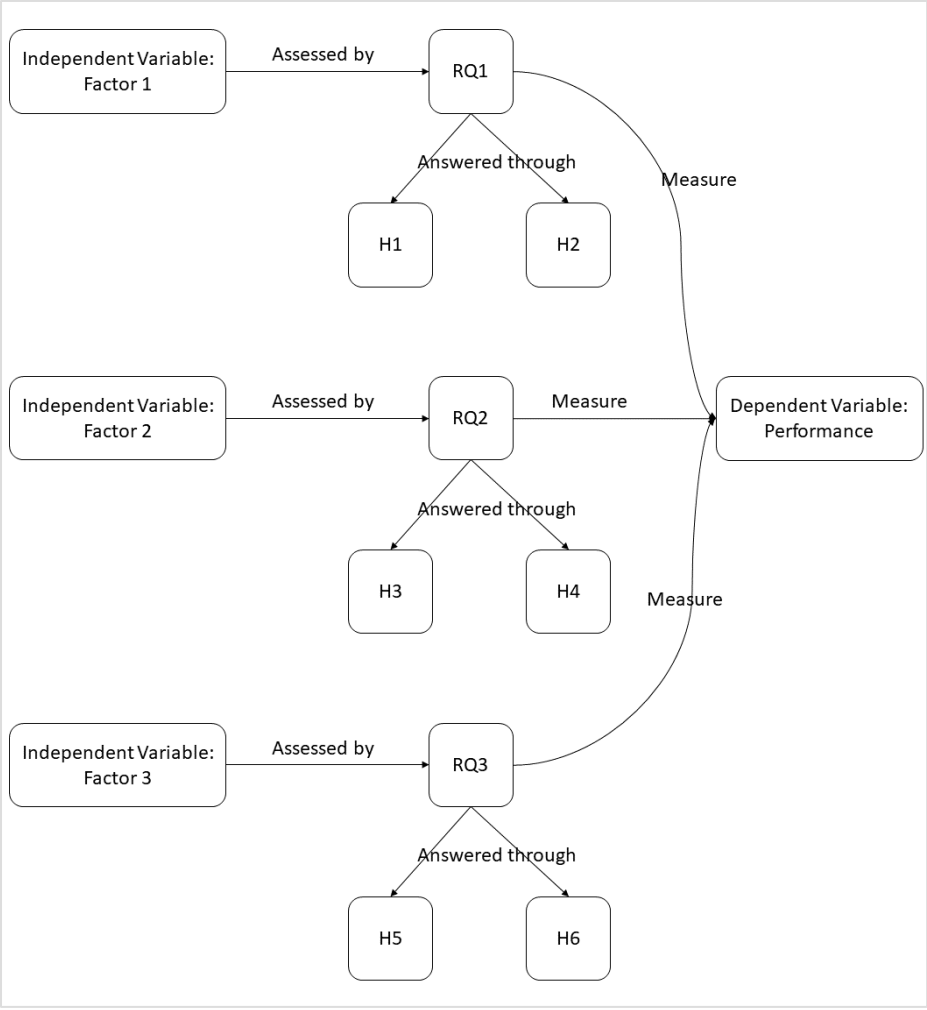


Figure 14. The Theoretical Framework of the Research

3.4. Conclusion

The research process is adopted from one DSR model. Following the methodology, the researcher has rephrased the solutions for the research problems several times since the research was initiated. For example, the technology to tackle the research problems was modified from general CNN architectures, ResNet and GoogLeNet, to the architectures specialised for object detection, SSD and YOLO, and eventually to two YOLO models. The RQs and hypotheses are amended with the solution changes. The RQs reflects the main factors in tackling the research problems. Hypotheses contain the candidate values of the factors, helping to plan and initiate the experiment, whose process is elaborated on in the next chapter.

4. Experimental Design

To verify all hypotheses mentioned in the last chapter, the author conducted an experimental project to collect data, whose process is detailed described in this chapter. The experimental design methods are explained in the first section. The second section outlines the experiment's process that includes three main tasks. The workflow or criteria in each task is illustrated in the three following sections.

4.1. Experimental Methods

The author designs the experiment by following the concept of factorial analysis (Yates, 1978). In a full factorial experiment, all possible values of experimental units (factors) are firstly listed. Next, the possible combinations of these values are included and cross-compared in the design, which is therefore called the fully crossed design. The design can show the effect of each factor and the interaction effects among factors on the response variable.

In contrast, the one-factor-at-a-time (OFAT) experiment examines the effect of a single variable and factor. Montgomery (2017) and Oehlert (2010) list the following advantages of the full factorial design compared with the OFAT design:

- Efficient
  Full factorial experiments can find optimal conditions faster than OFAT experiments. Besides, they give more information at the same or lower cost.
- Expandable
  Adding new factors in the experiments does not impose an additional cost in factorial experiments.
- Sensitive to the interaction effect
  When one factor's effect varies for different values of another factor, the OFAT experiment cannot detect the interaction. Factorial designs can illustrate how the response changes with the factors.

- Valid conclusion
  In factorial experiments, the effects of one factor are evaluated by considering all values of other factors. Therefore, the conclusion is valid over a range of experimental situations.

This experimental method is adopted for two reasons. Firstly, the research aims to provide highly accurate data to the bird preservation tasks. The fully crossed comparison can determine which combination of all factors' values can yield the best performance. The second proposal is to determine each factor's effect in the implementation. Future research can focus on tunning the factors with magnificent effect.

4.2. Design Overview

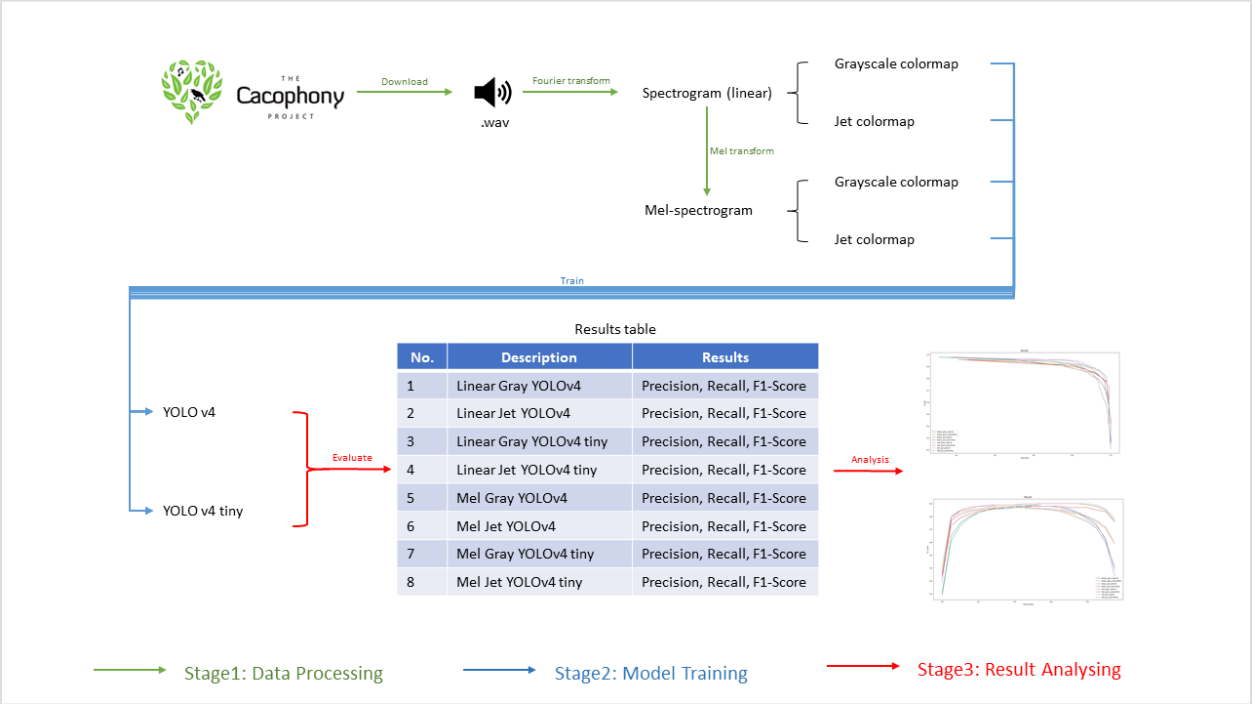
Following the factorial experiment design, the author firstly determines what factors are concluded in the experiments. Next, candidate values of the factors are selected based on their performance. As shown in section 2.4, three factors in the CNN process influence the solution's results. Each factor has several candidates with different performance in other research.

Linear spectrogram and Mel spectrogram are selected for the representation factor, as they show salient patterns of the morepork syllables that YOLO can recognise. About colourmaps, there were various options, such as "grey", "jet", "haline", and "balance" (Thyng, Greene, Hetland, Zimmerle, & DiMarco, 2016). The colourmap "grey" and "jet" were chosen to present the images because "grey" is the only colourmap with one channel, and "jet" covers all 8-digit colours among three-channel colourmaps. There are five series versions of YOLO architecture available. The 4th version was selected as the first four versions have a clear inheritance and upgrade relationship. In contrast, the so-called 5th version "has been frowned upon in the Computer Vision community" (Kanjee, 2020). The combination of all factors' values is listed in Table 4.

Table 4. Full Factorial Design

No.	Representation type	Colourmap	Architecture
1	Linear spectrogram	Grey	YOLOv4
2	Linear spectrogram	Grey	YOLOv4-tiny
3	Linear spectrogram	Jet	YOLOv4
4	Linear spectrogram	Jet	YOLOv4-tiny
5	Mel spectrogram	Grey	YOLOv4
6	Mel spectrogram	Grey	YOLOv4-tiny
7	Mel spectrogram	Jet	YOLOv4
8	Mel spectrogram	Jet	YOLOv4-tiny

As shown in Figure 15, the project's implementation process consists of three stages: data pre-processing, YOLO model training, and results analysing. In the data pre-processing step, all audio recordings are downloaded from the project's online database and eventually transformed into two spectrograms in two colourmaps. The second step trains two models by the four different presentations. Finally, the experiment yields eight sets of results, which are analysed, visualised in the third stage.



No.	Description	Results
1	Linear Gray YOLOv4	Precision, Recall, F1-Score
2	Linear Jet YOLOv4	Precision, Recall, F1-Score
3	Linear Gray YOLOv4 tiny	Precision, Recall, F1-Score
4	Linear Jet YOLOv4 tiny	Precision, Recall, F1-Score
5	Mel Gray YOLOv4	Precision, Recall, F1-Score
6	Mel Jet YOLOv4	Precision, Recall, F1-Score
7	Mel Gray YOLOv4 tiny	Precision, Recall, F1-Score
8	Mel Jet YOLOv4 tiny	Precision, Recall, F1-Score

Figure 15. The Overview of the Experimental Design

4.3. Stage 1: Data Processing

A Python program is created in this stage to retrieve the data and convert the data to YOLO format. The process includes processing audio recordings and their tags that locate the audios' bird syllables.

The procedure of processing audios can be divided into four steps (Figure 16). Firstly, the author downloads the recordings from the Cacophony Project database, followed by standardising all recordings into a single type. Next, the program converts recordings into four different types of images. Finally, all big images are cropped and sliced into small rectangle-shaped images.

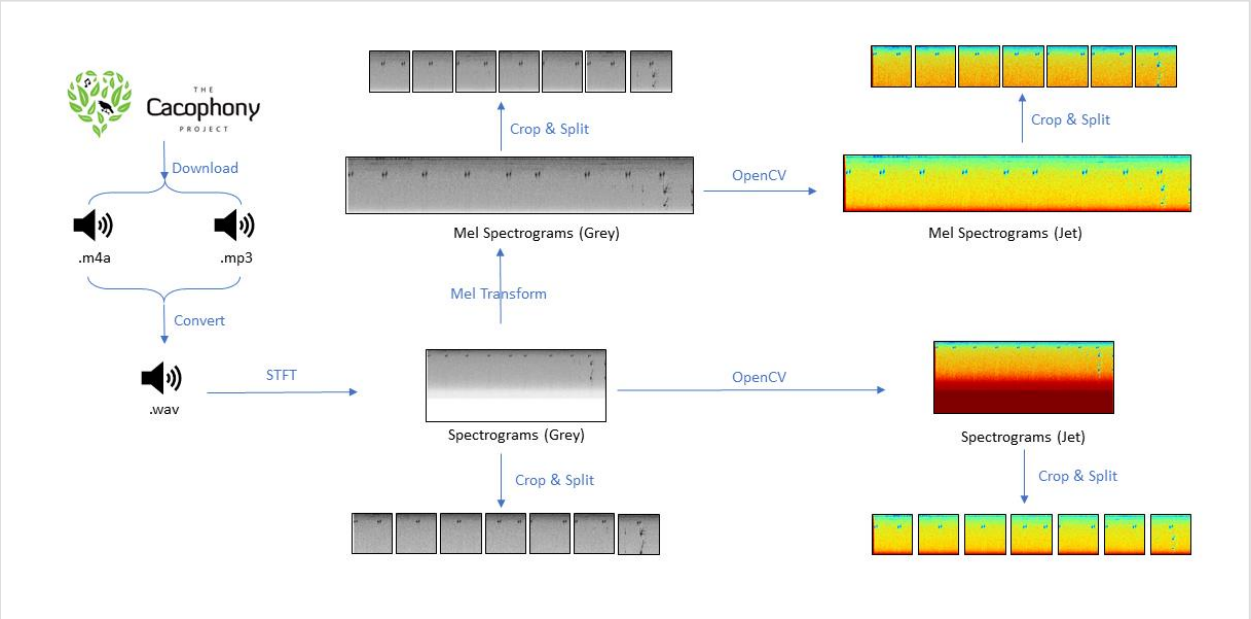


Figure 16. Processing Recordings Procedure

Figure 17 illustrates the steps of data processing, which localise every bird syllable in the sliced images. Firstly, the author retrieves a document that contains preliminary tags for the recordings from Dr Hunt. However, some tags in the document are irrelevant to this research, and not all syllables in audios are labelled. Therefore, the author revises and reconfirms all the tags and then transforms them into the YOLO format. Next, the program generates a tag file to localise all the syllables in each recording image. Finally, the large images are sliced into small images, and the program generated one tag file for every sliced image.

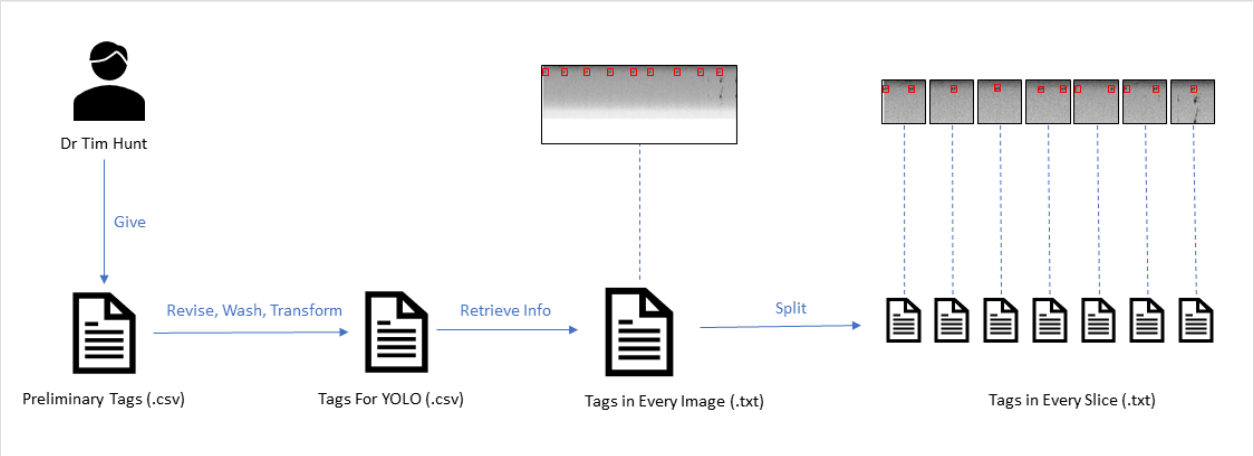


Figure 17. Processing Tags Procedure

4.3.1. Step 1: Download Audios

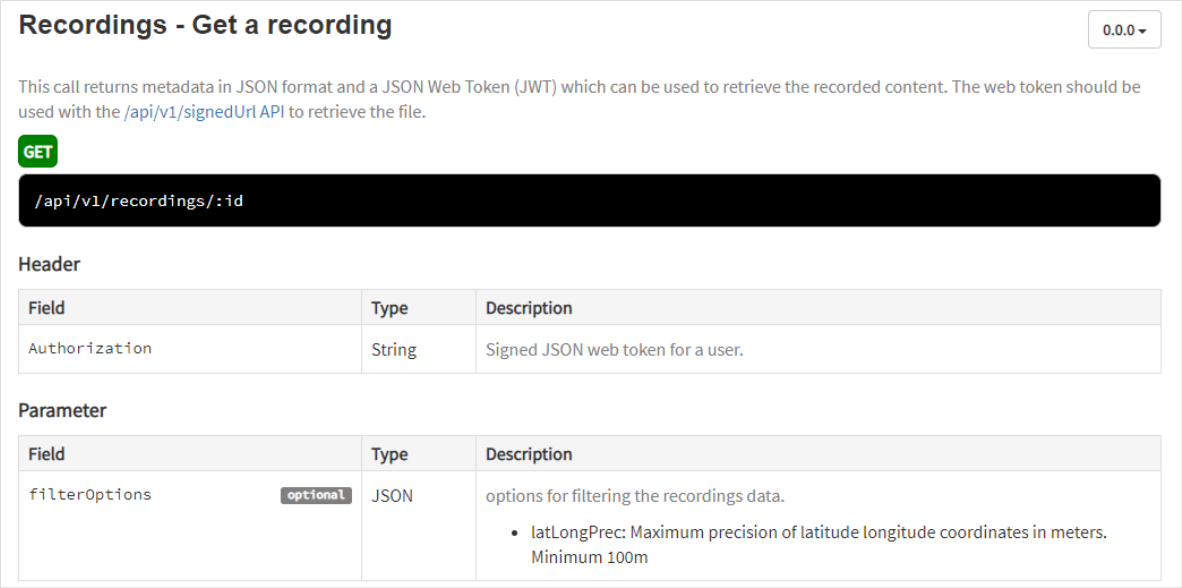
The Cacophony Project records bird audios by solar-powered devices in the forest and uploads them to its database (Hunt et al., 2019). Dr Tim Hunt is one manager of the project and responds to allocating authority to access the data. He provides the author with access to the recordings uploaded by four devices, whose information is listed in Table 5.

Table 5. Recorders' Information

Device ID	Device Name
378	fpF7B9AFNn6hvfVgdrJB
940	hammond_park_v2
1170	hammond_park_v3
1171	hammond_park_v4

Cacophony Project provides APIs (Figure 18) for public researchers to query and download recordings from its database. There is a pipeline in retrieving the audios. Firstly, the username and password are used to login into the server. After logging in, the program receives a user token used to query all recordings' information in available devices. By using the recording

information, the program gets downloading tokens from the server. The downloading tokens enable the program to download audio files to the local computer.



Recordings - Get a recording 0.0.0

This call returns metadata in JSON format and a JSON Web Token (JWT) which can be used to retrieve the recorded content. The web token should be used with the /api/v1/signedUrl API to retrieve the file.

GET

```
/api/v1/recordings/:id
```

Header

Field	Type	Description
Authorization	String	Signed JSON web token for a user.

Parameter

Field	Type	Description
filterOptions optional	JSON	options for filtering the recordings data. • latLongPrec: Maximum precision of latitude longitude coordinates in meters. Minimum 100m

Figure 18. The Cacophony API (The Cacophony Project, 2021)

The outcomes of this step are 1808 recordings downloaded from the Cacophony database. Each audio has a length ranging from 59 to 62 seconds. The recordings were recorded in the types of 'm4a' and 'mp3'. Dr Tim Hunt labelled these recordings by manually listening to them. According to the labels, only 656 of the downloaded audios contain morepork sounds. Therefore, these 656 recordings and their tags are selected for the next step.

4.3.2. Step 2: Convert Audios to Spectrograms

A spectrogram represents a signal's spectrum of frequencies varying with time (Almeida, 1994). A spectrogram can be created by repeatedly calculating the frequency of a short-time signal sample in a long time-domain signal to get a series of Fourier transformations. This transferring method is called "Short-time Fourier Transform", which is presented in Equation 1.

$$\mathbf{STFT}\{x(t)\}(\tau,\omega) \equiv X(\tau,\omega) = \int_{-\infty}^{\infty} x(t)w(t-\tau)e^{-i\omega t}\,dt$$

Equation 1. Short-time Fourier Transform Equation (Flanagan & Golden, 1966)

The transformation process contains four steps. Firstly, the audio file is loaded and sampled to audio time series using the sample rate of 22050. Next, the program imposes STFT calculation with the frame size of 2048 and the hop length of 512 to extract the matrix of short-term Fourier transform coefficients from the audio time series. The obtained coefficients matrix presents the spectrogram by complex numbers, which get powered to get real numbers. The real-number matrix is the linear spectrogram for the research. Then, the program maps the linear spectrograms directly onto the mel basis to get Mel spectrograms. The mapping function discards the frequencies higher than 6656 and uses the Mel channels of 416-channel. Finally, all gained linear spectrogram matrixes and Mel spectrogram matrixes are converted to decibel (dB) units, extended to the range from 0 to 255, and saved as images.

Each linear spectrogram image has a height of 1025 pixels and a width ranging from 2575 to 2664 pixels according to its audio's length. The Mel spectrogram image has the same width as its linear sibling while only has 416 pixels in height. Since every image uses only one number (channel) to describe its pixel's colour, these images are presented in greyscale.

4.3.3. Step 3: Present Spectrograms by Two Colourmaps

In this step, the researcher processes the one-channel greyscale images into three-channel Jet colourmap images. OpenCV, a real-time optimised computer vision library, is used to conduct the colourmap conversion. The different colourmaps of one audio's presentation are shown in Figure 19 and Figure 20.

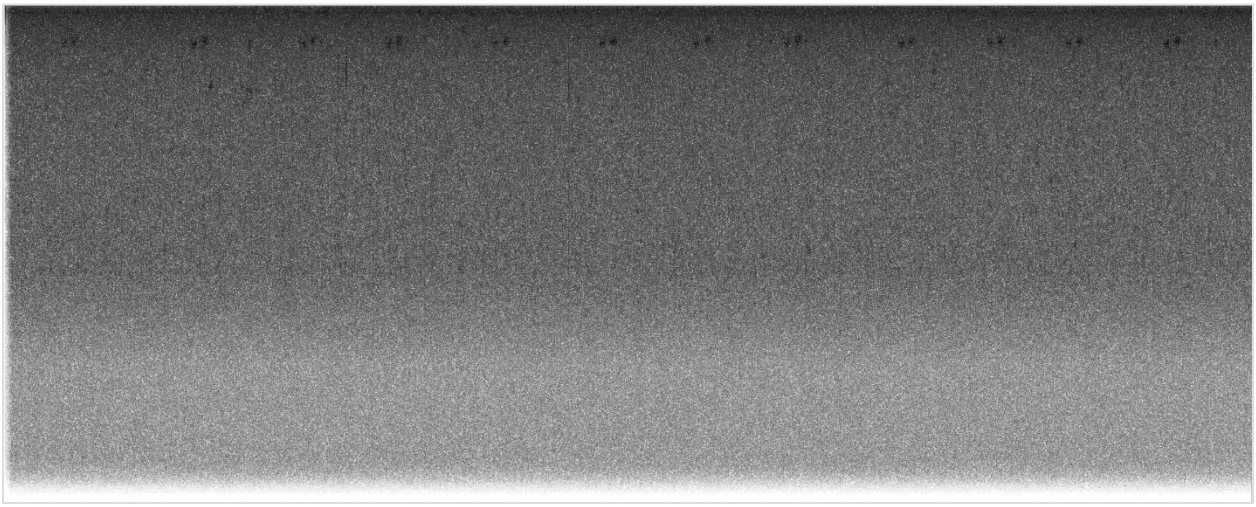


Figure 19. An Audio's Spectrogram in Greyscale Colourmap

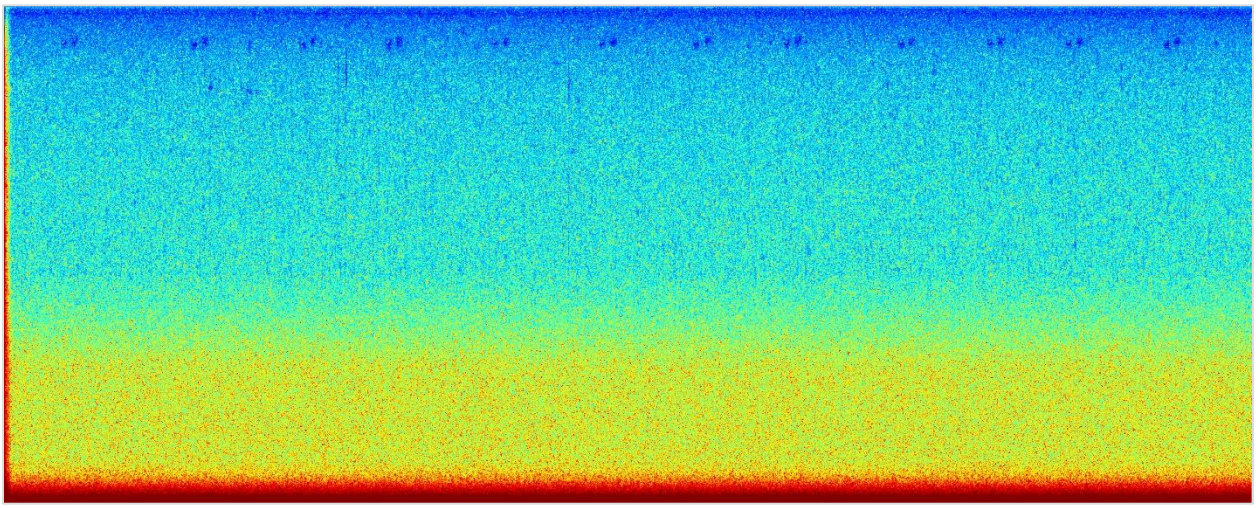


Figure 20. An Audio's Spectrogram in Jet Colourmap

4.3.4. Step 4: Split Images to Specific Size

The morepork sounds' frequency range is from 800 to 1100 (Hunt et al., 2019), which means the higher frequency area contains no valuable information and can be discarded. Besides, one particular image size that YOLO architecture requires is 416x416 (Alexey, 2020). Therefore, the author keeps the top 416 pixels of the images' y (height) axis and slices the images by 416 pixels with 60 pixels overlap in the x (width) axis.

The outcomes of this step are four folders of images. The four folds present the four combinations of two presentation types and two colourmaps. In each folder, there are 5248 images with the size of 416x416 pixels (Figure 21). The processed images are divided into two groups randomly: one group as the train data and the other group as the evaluation data.

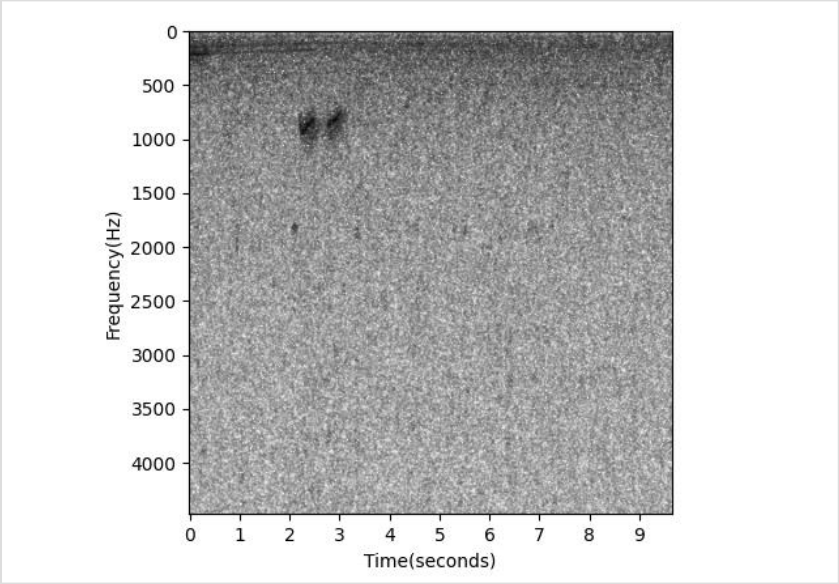


Figure 21. A Sliced Image (size: 416x416 pixel)

4.3.5. Step 5: Prepare Tags for Split Images

The original tags fetched from Dr Hunt are not perfect. The tags contain different categories such as "morepork", "maybe morepork", "ducks", "human", "other birds". The categories without the keyword "morepork" are irrelevant to this research, and therefore they need to be removed. The category containing the "maybe" keyword reflects that these tags have low confidence, requiring affirmation before being used. Besides, not all bird syllables are tagged out, and some tags are wrong.

All irrelevant tags are firstly removed in this step, leaving only the categories containing the "morepork" keyword. Then, the author verifies all chosen tags by listening to their audio and checking their spectrogram images. The verifying process ascertains that all bird syllables are

labelled and all tags are correct. Finally, the tags data is transformed to YOLO format, based on which one tag file is created for each spectrogram image.

4.4. Stage 2: YOLO Training

The Google Colab provides the runtime environment connected to the "Python 3 Google Compute Engine backend", with 12.69GB RAM and one Tesla T4 GPU with 15GB memory. The author saves the processed images and tag files on Google Cloud and uses these data to train YOLO models on Google Colab. The YOLO models implementation is based on the deep learning framework Darknet (Alexey, 2019).

Two architectures are adopted in this process, namely YOLOv4 and YOLOv4-tiny. YOLOv4 and YOLOv4-tiny share the same prediction stages with the same types of input, head, neck, and backbone (Bochkovskiy et al., 2020). The difference between these two architectures is the structure. The standard version has 162 floors with three floors making predictions, while the tiny version only has 38 floors with two floors making predictions (Alexey, 2020).

According to the design plan shown in Table 4, there are eight experiment cases, training two architectures by four folders of images. Each experiment case has a folder of 5248 images, 4461 (85%) chosen as the training data. The Darknet GitHub repository (Alexey, 2020) provides the framework for YOLO training and provides all settings of the YOLO family. Since the images with different representation types are presented in the same data structure, requiring no particular setting in the configuration. The two chosen colour maps have different matrix shape, and the two architectures require specific structure denotes. Therefore, four settings are required for these eight experimental cases.

Table 6 lists the common configurations in the setting files. All these four settings take the input images with both the width and height of 416. The "filters" at the layer before the YOLO layer is 18, and the "classes" at the YOLO layer is 1. The settings define the max train epochs for each model 4000. During the training epochs, the learning rate is $1.3x10^{-3}$ during the first 3200 steps,

$1.3x10^{-4}$ from 3200 to 3600 and $1.3x10^{-5}$ for the last epochs. Table 7 shows the different values in the configuration files. For cases that take greyscale images as input, "channels" and "hue" are set to 1 and 0, while for these take Jet images as input, the values are 3 and 0.1.

Table 6. Common Configurations in All Setting Files

Property	Value
width	416
height	416
filters (The layers before YOLO layers)	18
classes	1
max_batches	4000
steps	3200,3600
scales	.1,.1

Table 7. Differentiate Property Settings

No.	Description	Property	
		channels	hue
1	Grey YOLOv4 (Linear & Mel)	1	0
2	Grey YOLOv4-tiny (Linear & Mel)	1	0
3	Jet YOLOv4 (Linear & Mel)	3	0.1
4	Jet YOLOv4-tiny (Linear & Mel)	3	0.1

4.5. Stage 3: Result Analysing

The process of evaluating trained YOLO models contains three steps, as shown in Figure 22: 1) use the trained model to process the evaluating data to get predictions, 2) get results by comparing the predictions with the pre-labelled tags, and 3) visualise and evaluate all comparison results.

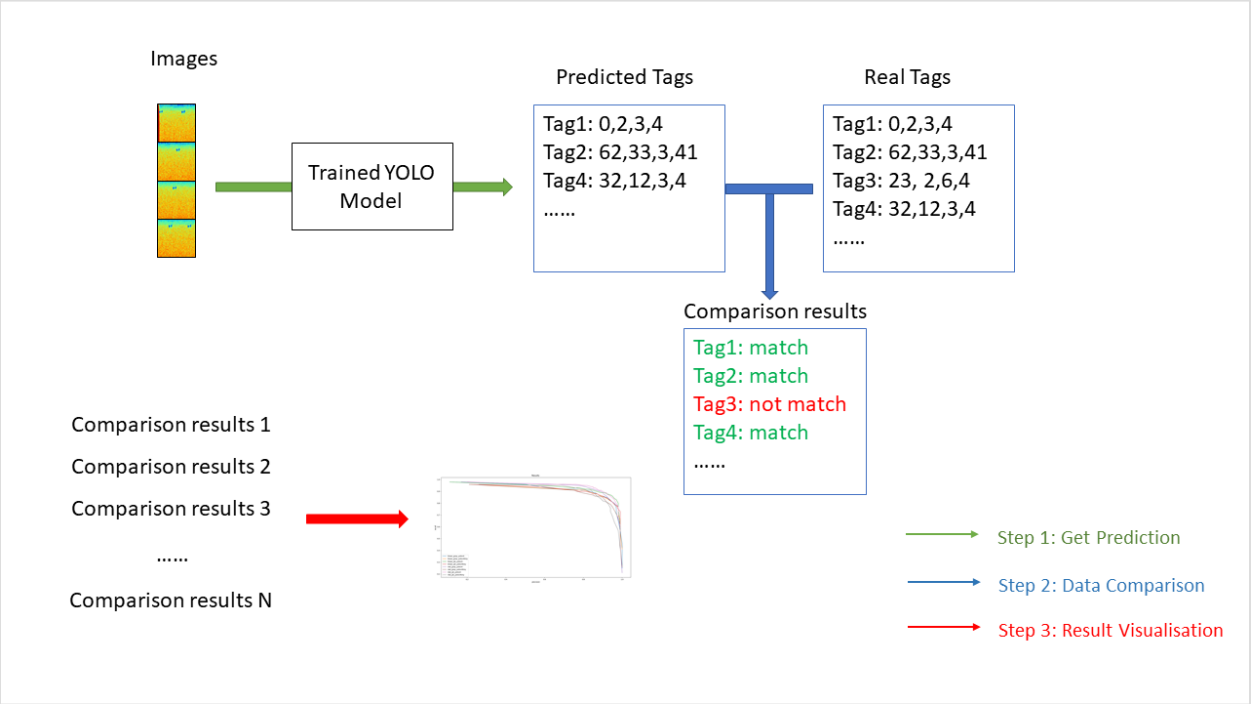


Figure 22. Results Analysis Process

4.5.1. Step 1: Get Predictions

Eight trained YOLO models are obtained from the training process. For each model, 787 (15%) of the processed images with 868 morepork tags are used as the evaluation data. As introduced in section 2.2.3, the YOLO model divides each image into grids and make multiple predictions in each grid. Each prediction contains a number, namely "score", to denote the confidence of a class in every bounding box. Before processing images and giving predictions, the models require the designations of the score threshold, which concept is explained in the next paragraph.

A YOLO model never returns predictions with a confidence score lower than the given score threshold (Redmon et al., 2016). For instance, when increasing the score threshold, the model needs more confidence to ascertain an object and make fewer predictions. This "cautious" operation guarantees a higher correct rate among all predictions, while at the same time, the model will miss the objects with low confidence. In other words, when adopting a larger score

threshold, the model's precision increases while its recall decreases. Since the models are anticipated to get a higher precision while not miss many tags, the author uses a range of score thresholds instead of a single threshold to discover models' best performance.

4.5.2. Step 2: Compare Predictions with the Ground Truth

In the previous step, the trained models predict bird syllable positions in the images. This step compares the predictions with the ground truth tags. The term "Intersection over Union" (IoU) is used in this step. IoU, also known as the Jaccard index (Jaccard, 1912), is a number to calculate the overlap percentage of a predicted box versus a grounding truth box for an object. Figure 23 shows the conception and the calculation of IoU. The IoU threshold makes YOLO models only return the predictions with a larger IoU value.

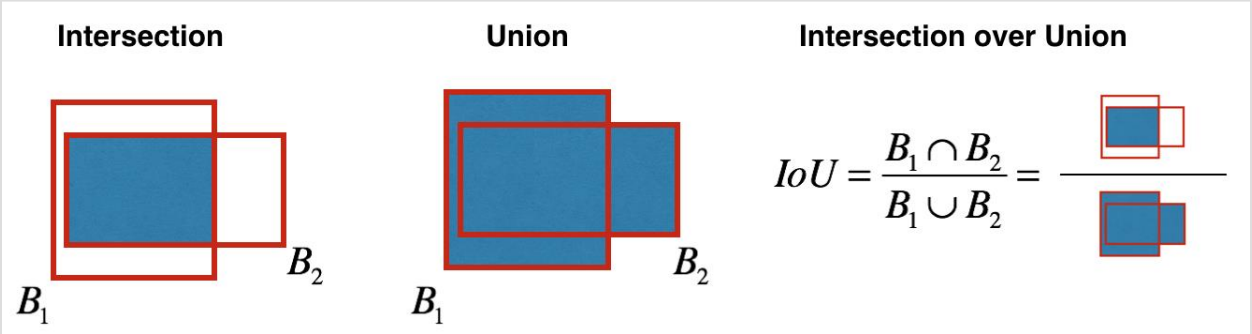


Figure 23. Intersection over Union (Oreilly, 2021)

By comparing retrieved predictions with the ground-truth labels, the author gets a confusion matrix. The confusion matrix contains the amount of true, false and missed prediction. From the confusion matrix, some terms such as precision, recall and F1 Score are calculated to indicate the performance of models. All terms involved in the step are listed as follows:

- Ground truth: A term refers to real or true information provided by direct observation and measurement.
- Confusion Matrix: A table with two columns and two rows that store true positives, false positives, true negatives, and false negatives.

- True Positive (TP): The predicted value and ground-truth value are both positive.
- True Negative (TN): The predicted value and ground-truth value are both negative.
- False Positive (FP): The predicted value is positive, while the ground-truth value is negative.
- False Negative (FN): The predicted value is negative, while the ground-truth value is positive.
- Precision: The percentage of correct positive predictions in all predicted samples, calculated by Equation 2.

$$Precision = \frac{TP}{TP + FP}$$

Equation 2. Precision (Moon et al., 2020)

- Recall: The percentage of correct positive predictions in all positive samples, calculated by Equation 3.

$$Recall = \frac{TP}{TP + FN}$$

Equation 3. Recall (Moon et al., 2020)

- F1 Score: The balance between precision and recall, calculated by Equation 4.

$$F1\ score = \frac{2}{Recall^{-1} + Precision^{-1}} = 2 \times \frac{Recall \times Precision}{Recall + Precision}$$

Equation 4. F1 Score (Moon et al., 2020)

- Precision-recall Curve (P-R curve): The line denotes different values of precision and recall by adopting different score thresholds.

- Average Precision (AP): AP is also known as the area under the P-R curve. Directly, as shown in Equation 5, the average precision covers all precision values when moving the recall value from 0 to 1. However, in practice and this research, the P-R curve is not a solid line but the connection of sparse dots. Therefore, the integral is approximated by a sum over the multiplying precisions by the change in the recall. The calculation is shown in Equation 6, where "N" is the total number that the image is split, "precision (k) " is the precision at the image's k fragment, and "**Δ**Recall(k) " is the change of recall between k-1and k fragment.

$$AP = \int_0^1 Precision(r)dr$$

Equation 5. The Strict Average Precision Calculation (WordPress, 2011)

$$AP = \sum_{k=1}^{N} Precision(k)\Delta\, Recall(k)$$

Equation 6. The Average Precision Calculation in Practice (WordPress, 2011)

In this prediction step, the author uses 19 different score thresholds increasing from 0.05 to 0.95 with a step of 0.05 for all eight trained models. The IoU threshold of 0.5 is used to verify the predicted tags. As a result, there generate 152 result files containing TP, FP, FN, precision, and recall.

4.5.3. Step 3: Analysis and Visualise Results

The result files have the terms and values in the log-type text, which must be collected and rewritten in an analysable format. The researcher uses the Python programming language and the "pandas" library to iterate all 152 files and retrieve performance terms and values. The retrieved values are reformed in a table and saved in eight CSV files.

Data visualisation gives readers an easier way to detect patterns, trends and frame in a complex dataset by placing data into maps or graphs (Heitzman, 2019). Rather than directly listing the tables and flooding readers in numbers, the author uses a set of techniques to extract meaningful information from the large group of results files and visualises the information in different types of graphs. The main tools used in the visualisation task are Python and its libraries, including "pandas", "numpy", and "matplotlib".

4.6. Evaluation Criteria

The literature review reveals that when researchers (He et al., 2016; Redmon et al., 2016; Simonyan & Zisserman, 2014) introduce new architectures, they highlight their models' two attributes: accuracy and speed. Therefore, the author creates evaluation criteria to assess and rank the eight experiment cases, covering the terms in categories of accuracy and speed.

As introduced in the earlier Section 4.1, the experiment is designed for two aims. One aim is an academic purpose that evaluates the influence of the three factors. The second aim is a practical purpose that determines the optimal combination of different factors' candidate values for real-world projects. Real-world AI projects can be categorised into real-time projects requiring immediate predictions and offline projects that analyse data in the database. The terms in the accuracy and speed categories are listed and weighed in the following sections.

4.6.1. Accuracy Terms

For the academic proposal, the research uses the AP term to denote a model's overall performance as it is the main evaluation metric used for object detection architectures. For example, all four YOLO versions' inventors use AP to statistically quantifies how good their architectures outperform existing models (Bochkovskiy et al., 2020; Redmon et al., 2016; Redmon & Farhadi, 2017, 2018).

For the practical proposal, the first accuracy term is the F1 Score. The bird preservation projects make decisions based on the quality of the available information (Bibby, 1999). Therefore, the more accurate predictions are, the better decisions the projects can make. In this research, high accuracy means missing the few birds' calls (high recall) and having few errors in predictions (high precision) simultaneously. Consequently, the F1 Score value, the harmonic mean of precision and recall, is taken to denotes how precisely a model can recognise a bird call. Using different thresholds, a YOLO model gives predictions with a range of precisions and recalls, by which a series of F1 Scores are generated. The real-world project requires a single threshold to make predictions. Thus, the best F1 Score a model can reach is the only consideration while its average performance can be neglected.

IoU is another accuracy term, measuring how precisely a model can localise the recognised syllable. Hunt et al. (2019) point out that decision-makers rely on counting bird call numbers for an extended period, which means the rightness of recognising a bird call is more important than the precision of localising the sound. Therefore, in these criteria, the recognition quality (F1 Score) overweighs the localisation quality (IoU) in the predictions.

4.6.2. Speed Terms

Reducing cost time in both training and executing process are research's directions. Many researchers, such as Simonyan and Zisserman (2014), He et al. (2016) and Redmon and Farhadi (2017), designed new architectures to increase not only models' accuracy but also their training

and execution speed. The training speed is not a considerate item in projects, as projects only utilise the already trained models. The difference in seconds or milliseconds in the execution process can be ignored by offline projects but is a critical factor for real-time applications. Overall, Table 8 presents all the terms and their weights in three categories.

Table 8. Terms and Weights in Two Types of Projects

Term	In Real-time Projects	In Not Real-time Projects
F1 Score	50%	80%
IoU	20%	20%
Executing Time	30%	0

Different terms' values are presented in different scale ranges and with various units; thus, the first step transforms the figures into the same scale as scores. Equation 7 presents the method to convert values to scores. In the formula, $\vec{z}$ means the term's value vector in all eight experimental cases, $\vec{z}_i$ denotes the value of an element at index i in the vector $\vec{z}$. Each term has a unique function, but the result is on the same scale as the term's score (TS), ranging from 0 to 100.

After the transformation, the array of values in different units are transformed to an array of numbers as the term's scores. Since every experimental case has five terms, the sum of five term's scores (TS_t) multiplied by its weight (TW_t) is taken as the experimental case's final score (Equation 8).

$$TS(\vec{z})_i = Function\,(\vec{z}_i) \quad (TS(\vec{z})_i \in [0,100])$$

Equation 7. Transfer Term Values to Scores

$$Case\ Score = \sum_t (TS_t * TW_t)$$

Equation 8. Experimental Case Score

4.7. Conclusion

This chapter presents the experimental design to collect data for this quantitative research. The design aims to ascertain the best combination of the three factors for real-world projects and determine each factor's influence on future research. The experiment adopts a full factorial design and forms eight cases to discover the effects and possible interactions of the three factors. This design uses fully crossed comparison to yields valid conclusions, while it costs less time than the OFAT method.

The results and findings from the experiments can be used in three sections, where each evaluating term has a different weight depending on the purpose. Evaluation criteria are therefore formed, proving a score standard and equations to assess all experiment cases. All experiment cases' results are listed in the next chapter.

5. Experimental Results

This chapter presents the result figures and the analysis from the figures. The results are categorised by five terms, namely F1 Score, IoU, AP, training time, and executing time.

5.1. AP

As discussed in section 4.5.2, when the score threshold increases, all models' precision increases while their recall decreases. Figure 24 shows the precision-recall curves of all models in the experience. All precisions maintain a high value in the figure when the recall is less than 0.8 but plummet rapidly after the recall overs 0.8. The AP denotes the average value of precision for recall values over 0 to 1. A more significant AP value means a model can get more precisive predictions among all recall and score threshold. Since AP also means the presentation under the precision-recall curves, it can be presented in the percentage number. For example, the figure 0.9476 in the chart is rewritten as 94.76% in the content.

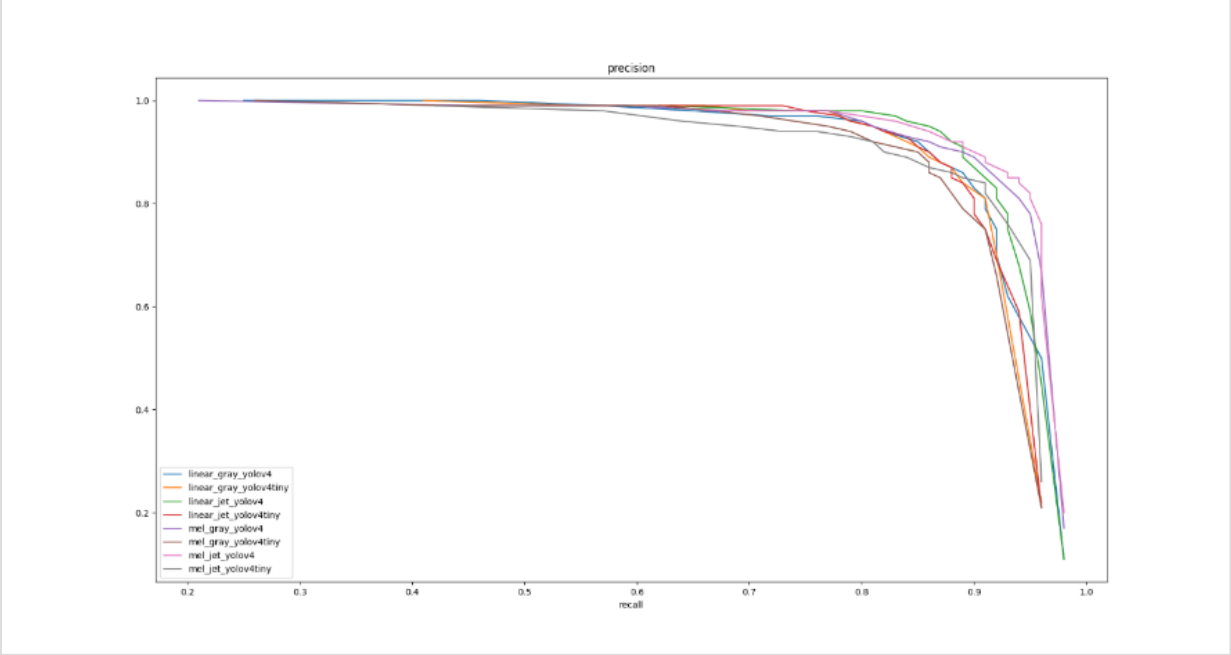


Figure 24. The Precision and Recall

All models' AP values are listed in the bar chart in Figure 25. The AP values are ranging between 91% and 95%. The values are grouped into three categories, the architecture shown in Figure 26, the spectrogram shown in Figure 27, and the colourmap shown in Figure 28, to compare these factors' influence.

In comparison, the architecture factor makes a magnificent difference to results. In All groups, the results achieved by YOLOv4 overperform the results obtained by YOLOv4tiny. On average, the AP of standard architectures is 94.02%, which is higher than that of tiny architecture (91.89%). It is noticeable that the smallest YOLOv4 AP value, 92.84%, is even higher than the largest YOLOv4tiny AP number, 92.3%.

Moving to different colourmaps, models achieve a slightly better AP by an average of 0.55% from the input of Jet images than greyscale images. The Jet colour map overperforms the greyscale in three of the four groups, with a difference of 1.16%, 0.29% and 0.89%, respectively. In the only group where the Jet underperforms the greyscale, the difference is only 0.13%.

Comparing the models' AP results grouped by spectrograms, there is no salient difference between the two groups. The deviation of two spectrograms' average values is negligible (0.25%). Besides, the two spectrograms are both in the lead in half of the groups.

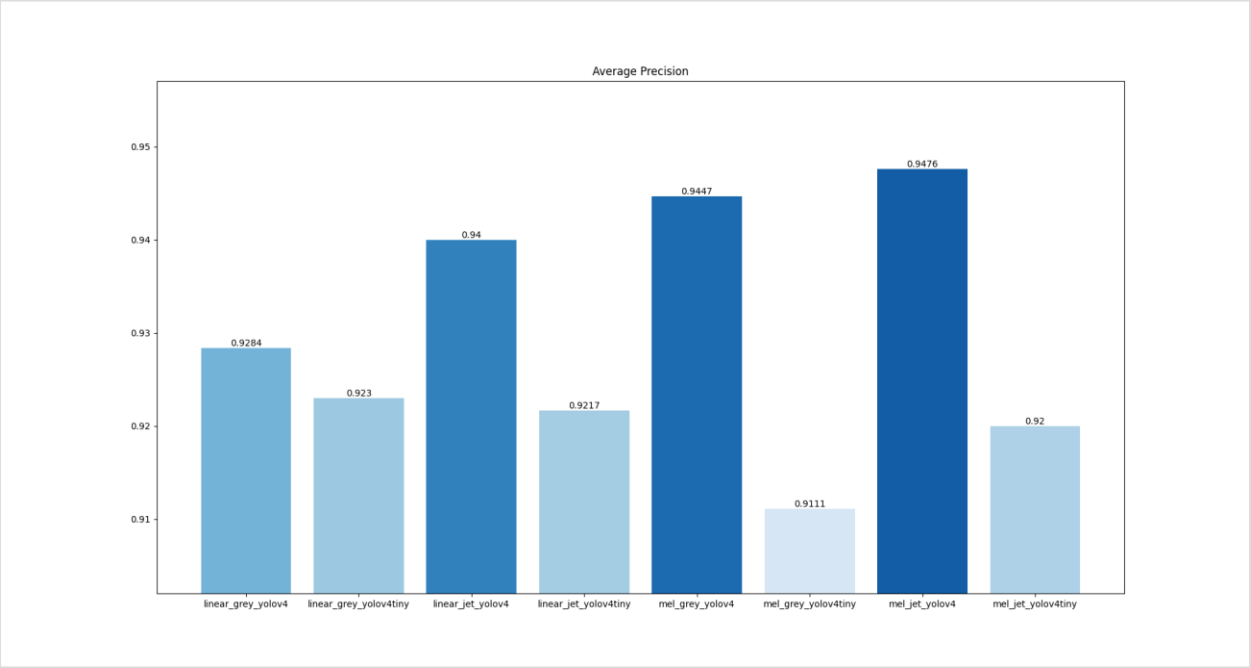


Figure 25. All Models' Average Precision

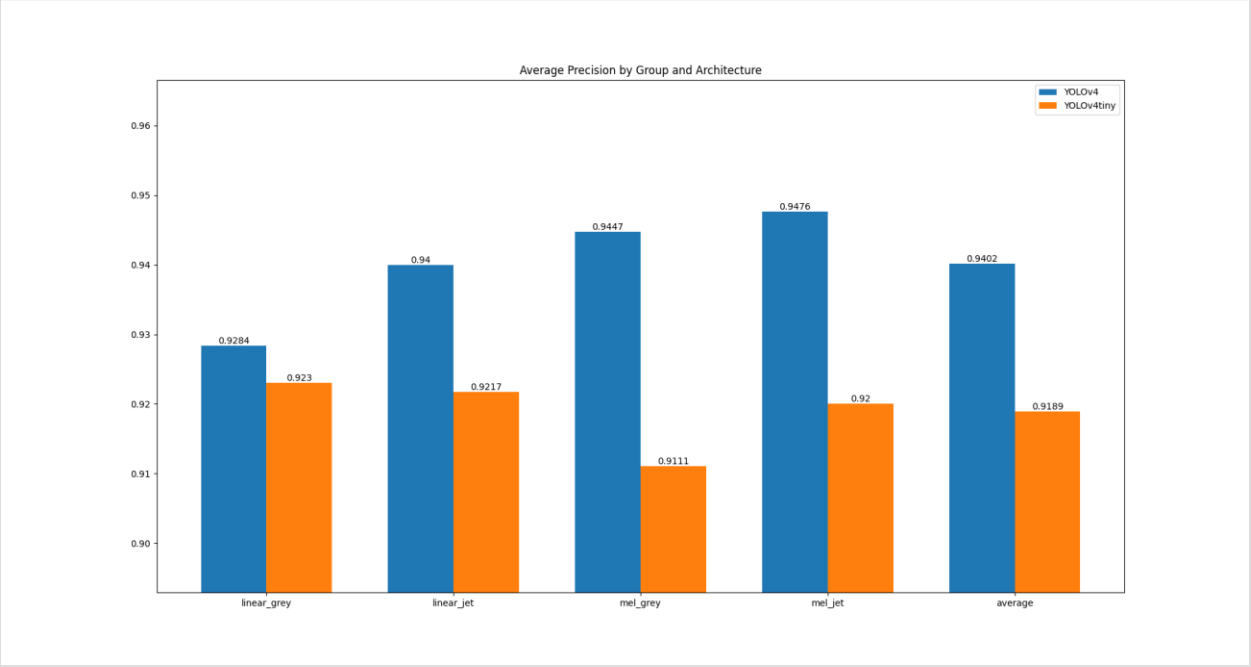


Figure 26. Models' AP by Group and Architecture

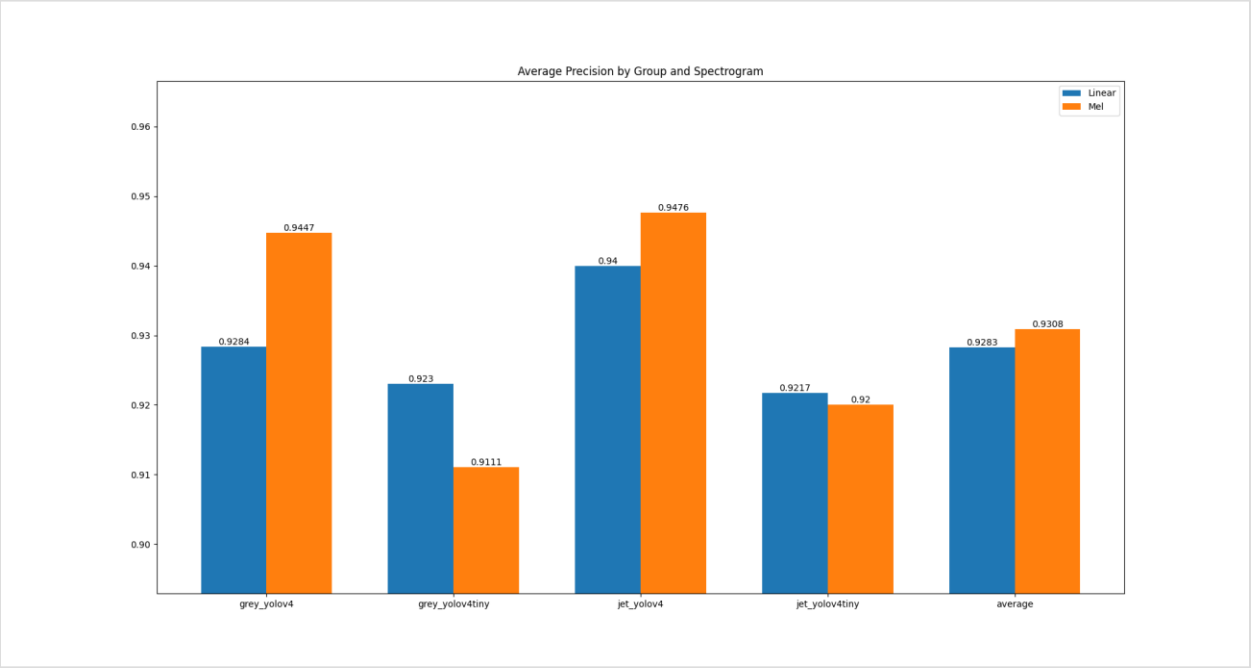


Figure 27. Models' AP by Group and Spectrogram

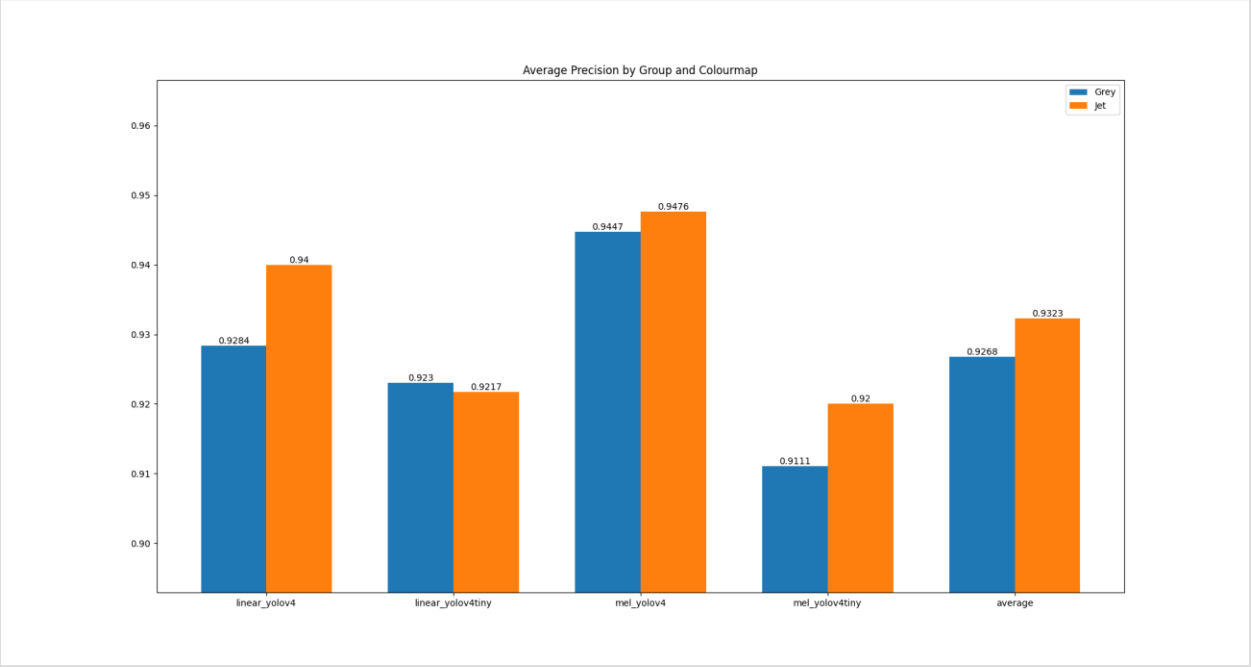


Figure 28. Models' AP by Group and Colourmap

5.2. F1 Score

Figure 29 and Figure 30 illustrate all models' precisions and recall over the sequential confidence threshold. At the confidence threshold near 0, all models have a recall higher than 0.95 but have a terrible low precision around 0.2. This data means the models give out predictions without high confidence. Thus, the predictions cover most of the bird syllables, but only 20% are correct. When the confidence threshold increases, the precision figures increase dramatically and then smoothly. On the other hand, the recall numbers drop slowly at the beginning and rapidly at the end. The trend map denotes that when the threshold is between 0.2 and 0.7, the models can obtain predictions with high precisions and recalls.

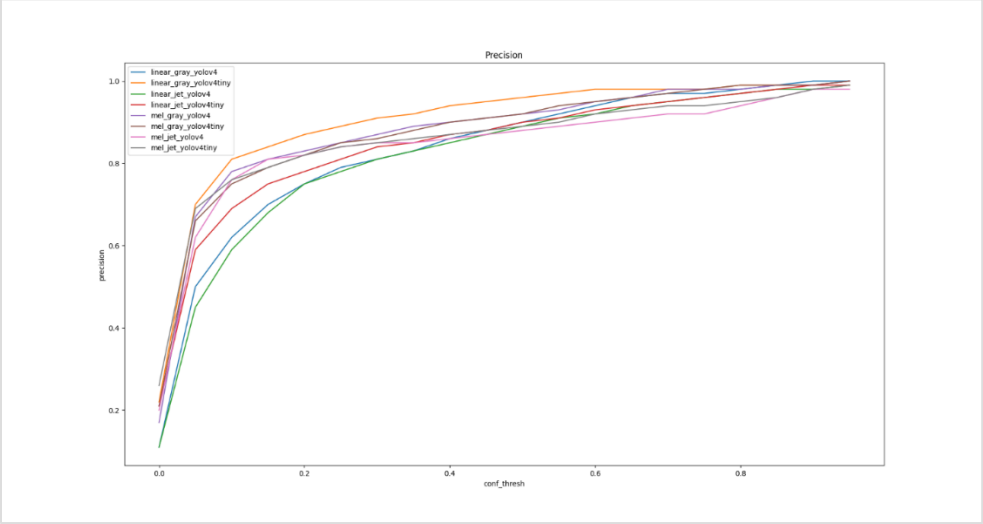


Figure 29. All Models' Precisions over the Whole Range Confidence Thresholds

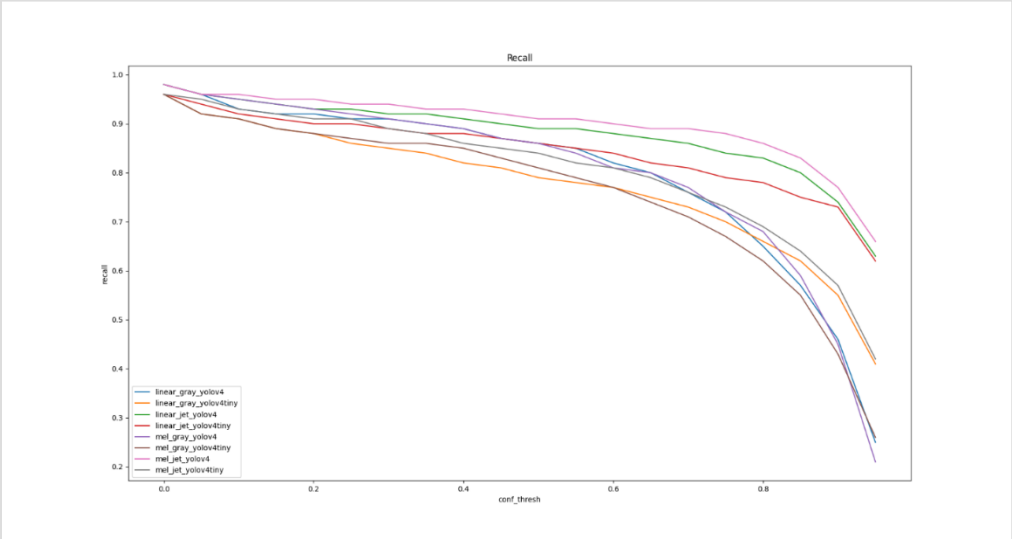


Figure 30. All Models' Recalls over the Whole Range Confidence Thresholds

Introduced in Session 4.5.2, the F1 Score is calculated by the precision and recall and denotes a model's performance in balanced considering the two figures. Figure 31 shows that all models' F1 Score in different confidence thresholds. When the threshold locates at the range from 0.3 to 0.55, all models obtain their highest F1 Score, ranging from 0.87 to 0.9. The combination of Mel spectrogram, Jet and YOLOv4, denoted by a pink line in the figure, shows the highest stability.

Figure 32 shows the highest F1 Score values that all models can reach in all thresholds. Two yolov4 models, combining with linear spectrogram & Jet colourmap and Mel spectrogram & Jet colourmap, achieve the most prominent figures of 0.9 among all cases. In contrast, the most diminutive figures are obtained by two yolov4 models.

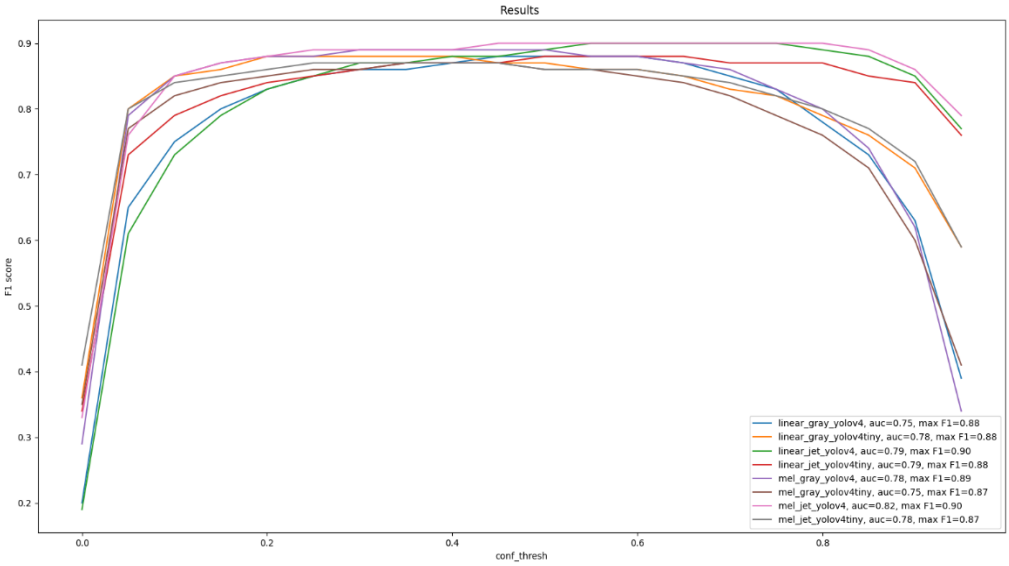


Figure 31. All Models' F1 Score over the Whole Range Confidence Thresholds

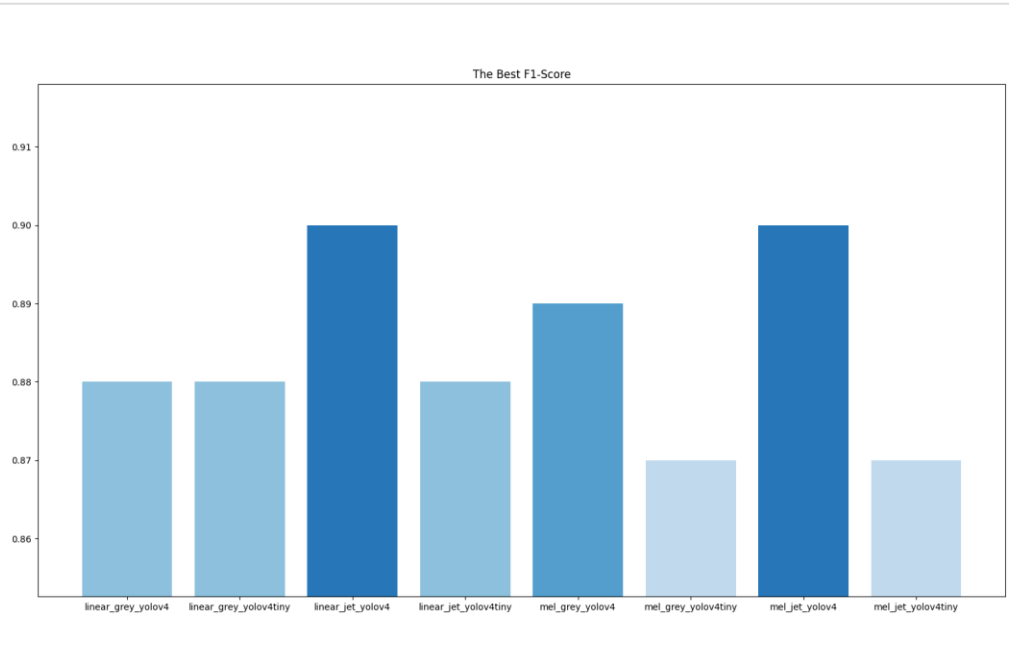


Figure 32. All Models' Best F1 Score

5.3. IoU

Introduced in Section 4.5.2, IoU is the number that denotes the overlapping percentage of a syllable's predicted box and its ground truth box. The higher this figure is, the more accurate a predicted box locates a bird syllable. Figure 33 illustrates the trend of average IoU among all confidence threshold range. The trend shows a salient positive association between IoU and the threshold. The experimental case of YOLOv4 architecture, mel spectrogram and grey colourmap, presented by the pink line, shows an overall highest average IoU among all cases. In contrast, the grey line is located at the bottom when the threshold is higher than 0.4. Other lines are entangled together without a clear pattern.

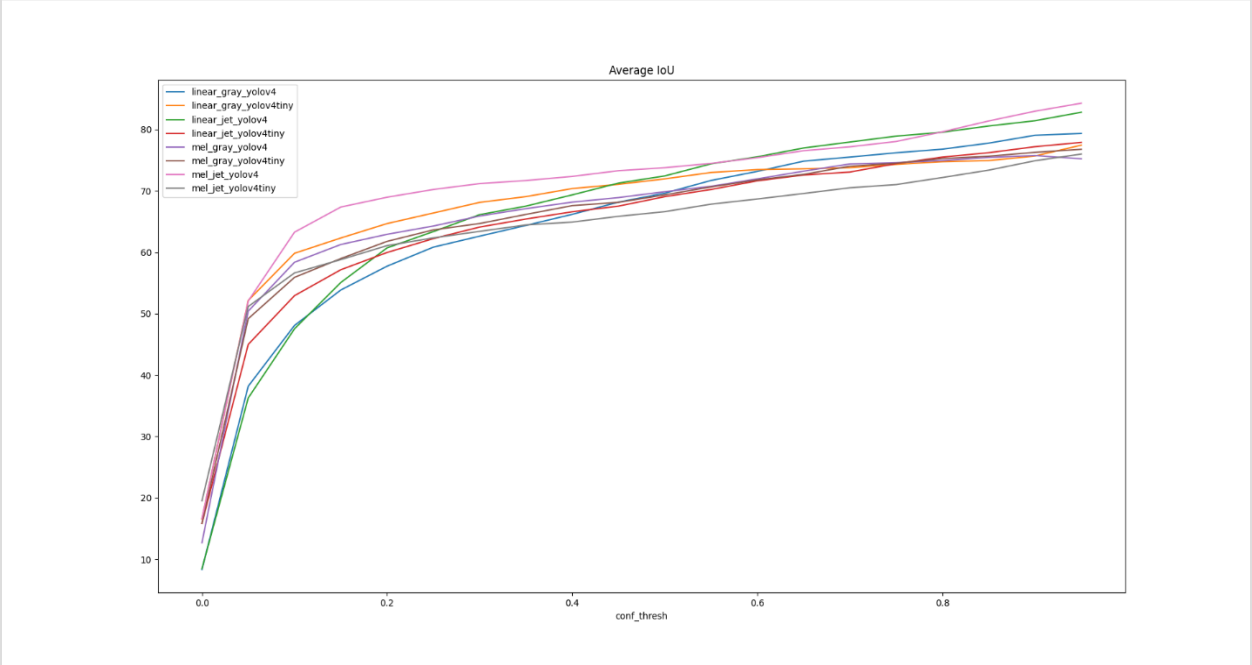


Figure 33. Models' Average IoU over Confidence Thresholds

Figure 34 shows the means of all models' average IoU. The 7th experiment case achieves an outstanding mean average IoU, 70.54%, while other cases have a number distributed in the range of 65%±2.2%. By comparing the results grouped by different factors, the figures show no evidence that one value overperforms the other value.

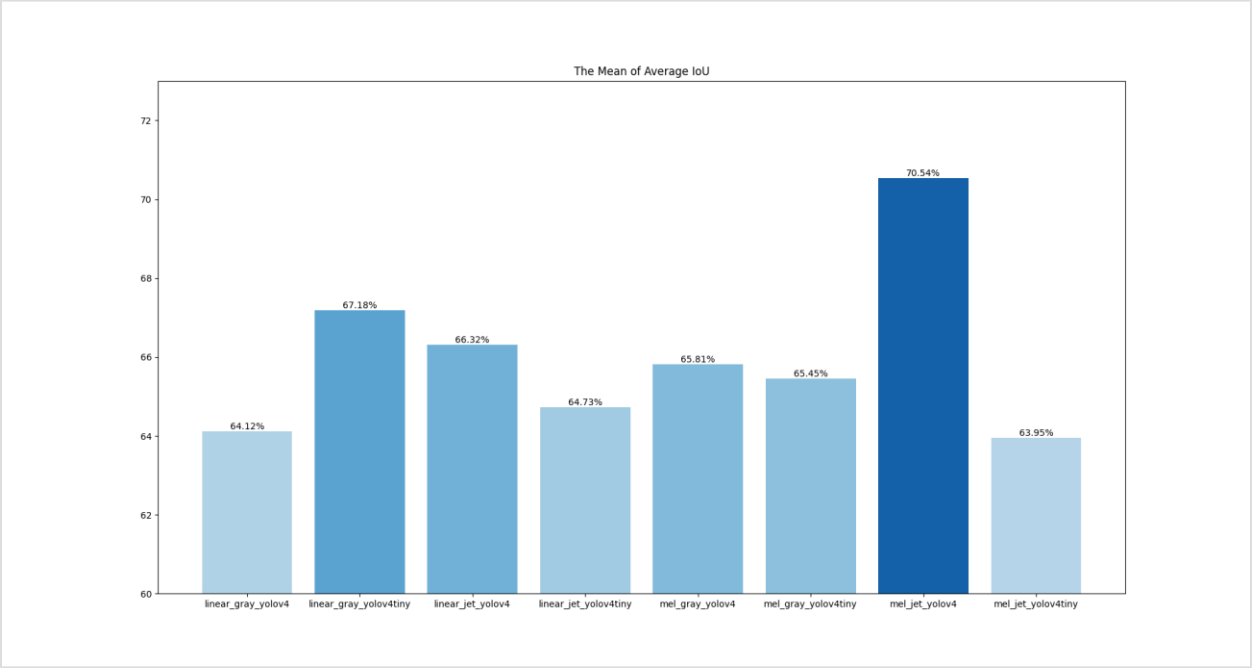


Figure 34. All Models' Mean Average IoU

5.4. Speed

Table 9 lists all the training time per epoch and executing time for one image during the experiments. The standard version architecture requires approximately eight times the training time and four times the execution time compared with the tiny version architecture. Training models by Jet map images costs around 2.5 times as by greyscale images, while their executing time is similar. The different time in different spectrograms' training and executing process is subtle.

Table 9. Models' Training and Executing Speed

No.	Combination	Training Time (seconds / epoch)	Executing Time (milliseconds / image)
1	Linear-Gray-v4	5.355	30.81
2	Linear-Gray-v4tiny	0.661	7.81
3	Linear-Jet-v4	9.249	30.50
4	Linear-Jet-v4tiny	1.578	8.39

No.	Combination	Training Time (seconds / epoch)	Executing Time (milliseconds / image)
5	Mel-Gray-v4	5.357	31.32
6	Mel-Gray-v4tiny	0.655	7.94
7	Mel-Jet-v4	9.252	34.18
8	Mel-Jet-v4tiny	1.573	8.39

5.5. Evaluation Scores

The outcomes from the transformation are evaluation scores of the three terms in the eight experiment cases. The total scores for two types of projects are calculated by applying Equation 8. The scores of individual terms and total sum are presented in Table 10. For real-time projects, the second case combining the linear spectrogram, grey colourmap and YOLOv4 tiny architecture achieve the highest score. For not real-time projects, the YOLOv4 architecture using mel spectrogram and jet colourmaps obtains the best score.

Table 10. The Evaluation Score

No.	Combination	F1 Score	IoU	Executing Time	Total (Real-time projects)	Total (Not Real-time projects)
1	Linear-Grey-v4	88	64.12	53	72.724	83.224
2	Linear-Grey-v4tiny	88	67.18	88	83.836	83.836
3	Linear-Jet-v4	90	66.32	53	74.164	85.264
4	Linear-Jet-v4tiny	88	64.73	87	83.046	83.346
5	Mel-Grey-v4	89	65.81	52	73.262	84.362
6	Mel-Grey-v4tiny	87	65.45	88	82.99	82.69
7	Mel-Jet-v4	90	70.54	48	73.508	86.108
8	Mel-Jet-v4tiny	87	63.95	87	82.39	82.39

5.6. Conclusion

There are various ways to access the performance of a neural network. Each method reflects one or some of the models' aspects, such as the predictions' positive and negative rate, overall positive correctness rate for various threshold values, and executing time. The author chooses

five evaluation terms to illustrate the terms' results. The AP charts reflect the overall accuracy of the models. The AP figures grouped into different categories reveal the performance differentiation caused by each factor. The F1 Score and IoU charts show the correct rate of how models recognise and localise the syllables in the images. The cost time of all cases is also listed to denote the models' speed in the training and executing process. Finally, eight models' overall evaluation scores are calculated by following the evaluation criteria. The findings from the calculation and comparison are presented in the next chapter.

6. Discussion

This chapter discusses the research's conformation to the research design described in Chapter 3. The hypotheses of the three research questions are firstly identified with research evidence. Next, the author discusses the potential reasons behind the experiment results based on the literature review and personal understandings.

6.1. Research and Research Design

This section relists the three research questions and their hypotheses. Table 11 summarises all hypotheses' validations supported by research evidence.

Table 11. Hypotheses' Validation and Research Evidence

Variable	Research Question	Hypothesis	Research Evidence
Representation type	RQ1	H1 – Disproved H2 – Disproved	1. As discussed in Section 5.1. 2. Figure 27 illustrates that the variance caused by different audio representation types is negligible.
Colourmap type	RQ2	H3 – Validated H4 – Validated	1. As discussed in Section 5.1. 2. Figure 28 illustrates that the variance caused by different representation colourmap types is noticeable.
Architecture	RQ3	H5 – Validated H6 – Validated	1. As discussed in Section 5.1. 2. Figure 26 illustrates that the variance caused by different YOLO architectures is magnificent.

The following three blocks list the detailed discussions of the research questions.

6.1.1. Research Question 1

Figure 35 shows the mapping of RQ1, hypotheses, as well as the variables in the research:

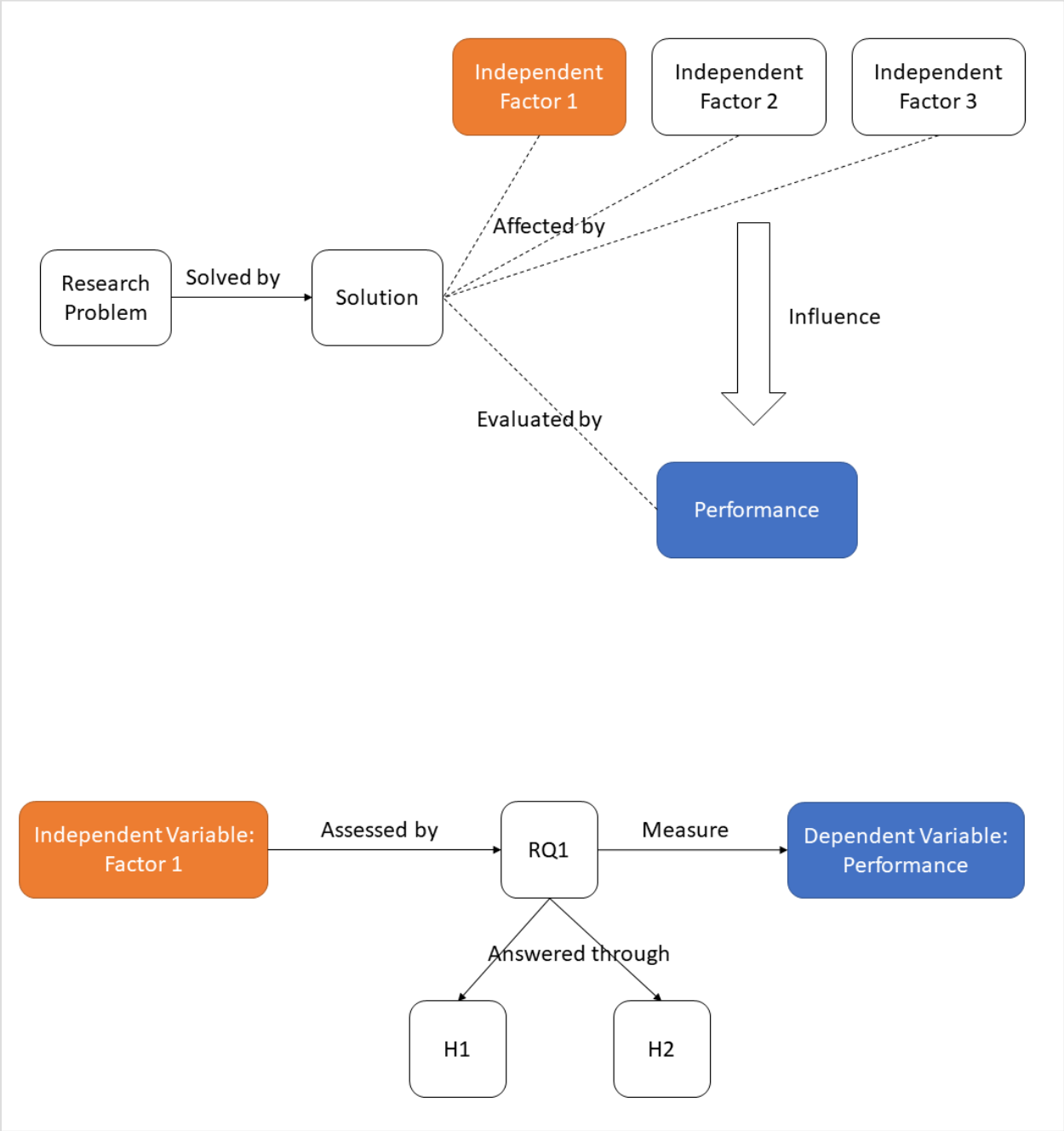


Figure 35. The Mapping of RQ1

Variables, RQ1, H1 and H2 are re-stated as follows:

Independent Variable Factor 1: Spectrogram

Dependent Variable: The performance of using YOLO in detecting morepork sounds

RQ1: Does the audio representation type influence the performance of using YOLO in detecting morepork sounds?

H1: Spectrogram impacts the performance of using YOLO in detecting Morepork sounds.

H2: Between two spectrograms, linear spectrogram and Mel spectrogram, the Mel spectrogram makes YOLO achieves better performance in detecting Morepork sounds.

Reflection:

The H1 is a simple hypothesis to assume the relationship between audio representation type and a model's performance. The H2 is a directional hypothesis constructed from the literature review to predict the optimal audio representation type for the performance. H1 and H2 are both proven false by the experimental results.

Research Evidence:

In the eight experiment cases, four adopts the linear spectrogram, and four adopts the mel spectrogram. As discussed in section 5.1 and Figure 27, the two groups' mean average precisions are similar, and they both have a leading performance in half of the comparisons.

6.1.2. Research Question 2

Figure 36 shows the mapping of RQ2, hypotheses, as well as the variables in the research:

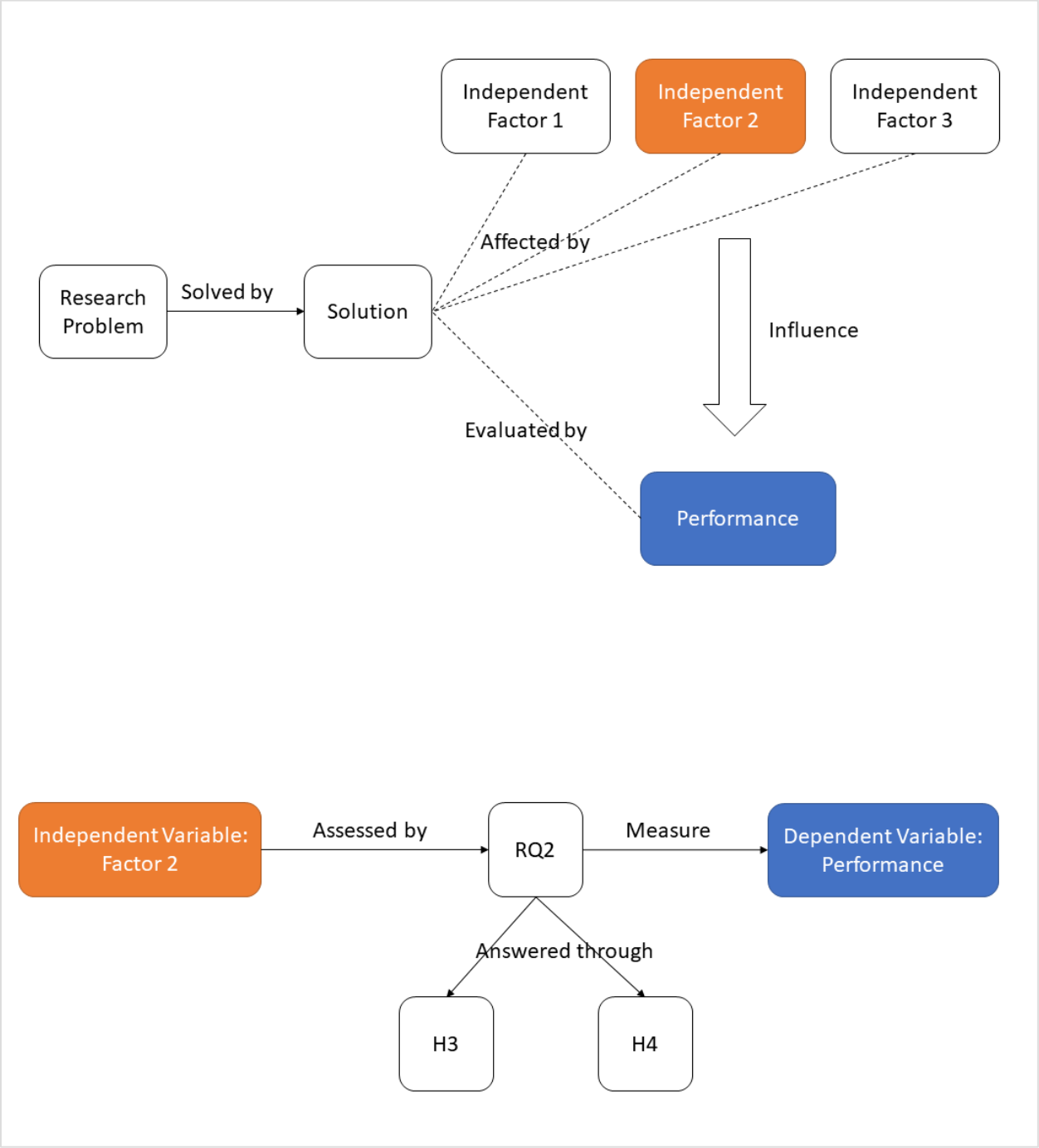


Figure 36. The Mapping of RQ2

Variables, RQ2, H3 and H4 are re-stated as follows:

Independent Variable Factor 2: Colourmap

Dependent Variable: The performance of using YOLO in detecting morepork sounds

RQ2: Does the colourmap influence the performance of using YOLO in detecting morepork sounds?

H3: Colourmap impacts the performance of using YOLO in detecting Morepork sounds.

H4: Between two colourmaps, Greyscale and Jet, the Jet colourmap makes YOLO achieves better performance in detecting Morepork sounds.

Reflection:

The H3 is a simple hypothesis to assume the relationship between audio representation colourmap and a model's performance. The H4 is a directional hypothesis constructed from the literature review to predict the optimal audio representation colourmap for the performance. H3 and H4 are proven true by the experimental results.

Research Evidence:

In the eight experiment cases, four adopts the Jet colourmap, and four adopts the greyscale colourmap. As discussed in section 5.1 and Figure 28, the mean average precisions of Jet colourmap surpasses that of the greyscale colourmap. In four group comparisons, three groups achieve a higher mean AP by Jet colourmap while the last shows similar results.

6.1.3. Research Question 3

Figure 37 shows the mapping of RQ3, hypotheses, as well as the variables in the research:

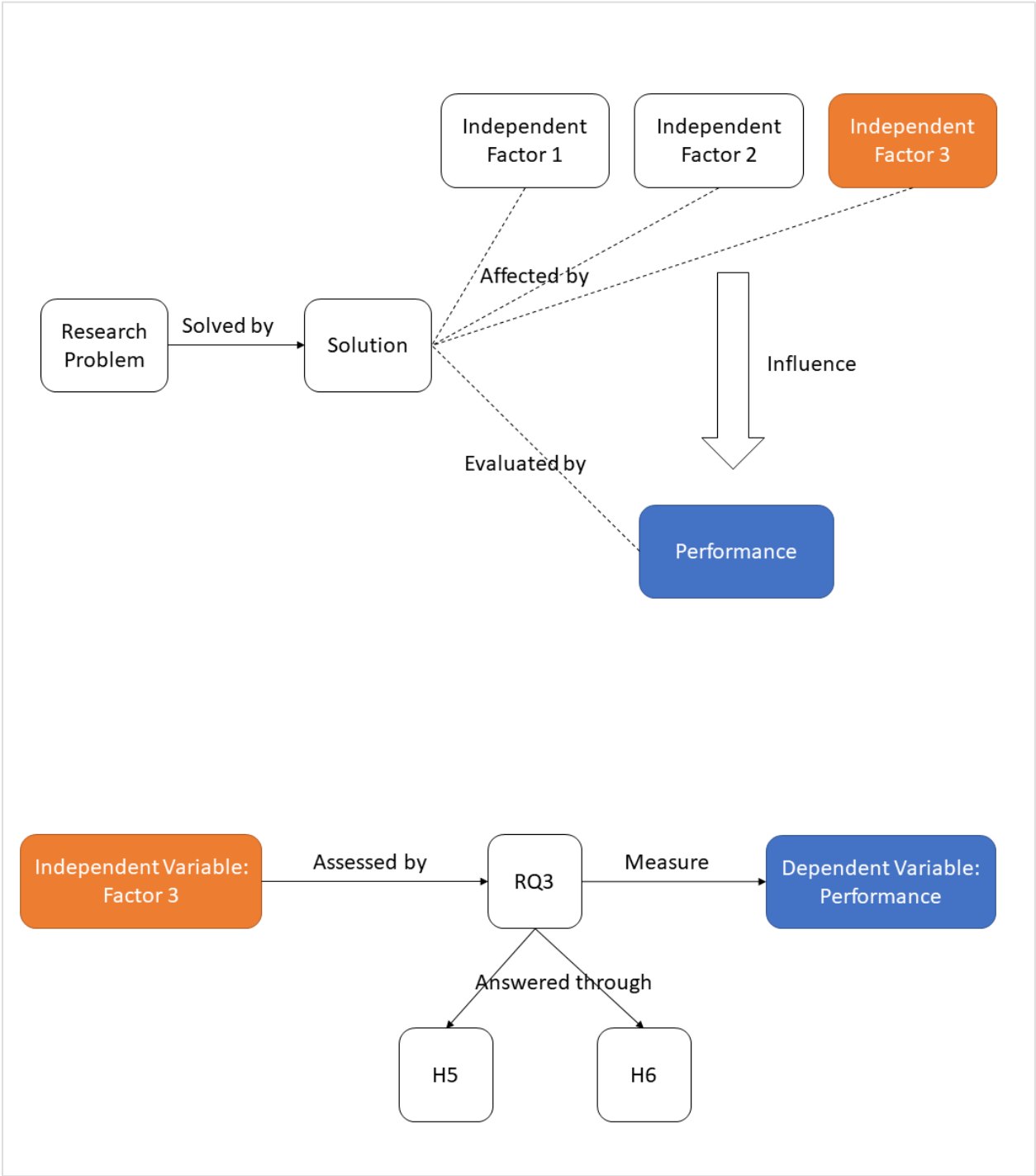


Figure 37. The Mapping of RQ3

Variables, RQ3, H5 and H6 are re-stated as follows:

Independent Variable Factor 3: Architecture
Dependent Variable: The performance of using YOLO in detecting morepork sounds
RQ3: Does the architecture influence the performance of using YOLO in detecting morepork sounds?
H5: Architecture impacts the performance of using YOLO in detecting Morepork sounds.
H6: Between two architectures, YOLOv4 and YOLOv4-tiny, the YOLOv4 architecture makes YOLO achieves better performance in detecting Morepork sounds.
Reflection:
The H5 is a simple hypothesis to assume the relationship between audio representation type and a model's performance. The H6 is a directional hypothesis constructed from the literature review to predict the optimal audio representation type for the performance. H5 and H6 are both proven truth by the experimental results.
Research Evidence:
In the eight experiment cases, four adopts the YOLOv4 architecture, and four adopts the YOLOv4 tiny architecture. As discussed in section 5.1 and Figure 26, the mean AP of YOLOv4 surpasses that of YOLOv4 tiny. In all four group comparisons, YOLOv4 achieve a higher AP than the YOLOv4 tiny cases.

6.2. Conclusion

Based on the listed figures and charts in Chapter 5, all six hypotheses of three RQ are verified. Among the three factors, only the architecture makes a salient difference. The YOLOv4 tiny model is created based on the standard YOLOv4 architecture (Jiang, Zhao, Li, & Jia, 2020). Compared with the standard version, the tiny model used in the experiments has fewer layers (29 vs 137) and fewer parameters (18.8 MB vs 162 MB) and therefore requires fewer memories and GPU capacity. The tiny version makes a sacrifice on the accuracy but can be trained and run at a much faster speed. The author trained these two models with the same dataset on Google Colab

with the "NVIDIA-SMI 460.67" graphic card. The training process was approximately 8 hours for the standard version but was only 43 mins for the tiny version.

Choosing different colourmaps and spectrogram types aims to give the model different feature presentations to recognise. However, in this experiment, the colourmaps affect the results slightly, and the spectrogram types do not make a difference. The reason is that all test cases reach the best results for the one-class object detection, and features of different colourmaps and presentation types are not salient enough to sway the results. The two factors should be evaluated in the dataset with ten or more classes.

Another factor that should be mentioned is data quality. After getting the tags from Dr Hunt, the author used the tags directly without reviewing them. As a result, the F1 Score was disappointingly low at 0.65. After scrutinising and correcting the labels one by one, the author obtained a much better F1 Score above 0.8. Due to this experience, the author would suggest: When using YOLO in single or double-class objects detecting tasks, engineers should pay most effect in selecting the architectures and verifying the training data quality.

For not real-time projects that process data in the database and not sensible to the speed, such as Cacophony Project, the best option is using YOLOv4 architecture with Mel spectrograms and Jet colourmaps. This option gives the best accurate predictions, though it requires more time to train and execute than all other options. On the other hand, for real-time projects that require immediate predictions, the best option is using YOLOv4-tiny architecture with linear spectrograms and grey colourmaps. The option sacrifices some accuracy but still gives promising results in the shortest time.

7. Conclusion

This research aims to solve a real-world problem, providing accurate information for the Morepork preservation projects. Compared with the current approach (Hunt et al., 2019) in the morepork detection task, which achieves the F1 Score of approximate 0.75, the YOLO technique obtains a much better performance with an F1 Score of 0.9.

In addition to the highly accurate recognitions, the YOLO architecture can precisely localise the sound syllables in the spectrogram image. The best IoU the experimental models achieved is 70.54%, while others are distributed in the range of 65%±2.2%. The IoU in the experimental results contains offsets in two dimensions, the time axis and frequency axis. If only take the time axis into the calculation, the IoU would be much higher than the obtained number. Luckily, projects only require localising syllable in a specific time range; Therefore, the models can localise the bird calls for projects with the IoU higher than 70.54%.

Different YOLO architectures obtain noticeable difference in results; the YOLOv4 models have the AP ranging from 92.84% to 94.76%, while the AP of YOLOv4 tiny models is between 91.11% and 92.3%. As a price, the standard version costs eight times the training time and four times the execution time compared with the tiny version. The colourmap type also makes a variation in the results. On average, the Jet colourmap overperforms the greyscale colourmap slightly by 0.55%. Since the Jet colourmap images contain three channels while the greyscale images have one channel, Jet colourmap requires more parameters in the model and takes more time in training. The third factor, spectrogram type, makes no difference in the performance or speed.

7.1. Research Limitations

CNN models' performance drops when they have more classes. This experiment is designed for only one class, Morepork sounds. Therefore, all experimental cases can reach the ceiling accuracy that YOLOv4 architectures can obtain, near 95% (Bochkovskiy et al., 2020). In this

circumstance, the difference between two colourmap values and two spectrogram types are insignificant. Incze's (2018) experiment shows a similar conclusion: he has a minor accuracy difference (2%) in the 2-class tests but obtains 8% and 10% gaps in the ten and 50-class tests, respectively.

Besides, the evaluated architectures are limited in the YOLOv4 family. The YOLOv4 tiny architecture is originated from the standard version, with shallower layers and fewer parameters. Therefore, adopting the YOLOv4 and YOLOv4 tiny models give two options for the bird preservation projects but has little academic magnificence.

The reason for using only one architecture series is the shortage of time. This research is used as the author's master project and required to be finished in one semester. The research process took five months, with a one-month system literature review and study, two-month experimental implementation and two-month writing this report. To understand, implement and assess another architecture, the author requires at least two more months.

7.2. Future Research

For further research, more bird species will be involved. An experiment with 10 to 20 bird species is anticipated to distinguish result differentiation among the groups using different factor values.

Another object detection framework is Single Shot MultiBox Detector (SSD) (Liu et al., 2016). SSD uses a single deep neural network to process an input image or a video source to detect objects. Like YOLO, SSD also gives bounding boxes to localize objects and objects' classes. Comparing the object detection performance between SSD and YOLO is a hot topic. Researchers compare their accuracy and speed in detecting objects such as outdoor urban advertising panels (Morera, Sánchez, Moreno, Sappa, & Vélez, 2020), agricultural greenhouses (Li, Zhang, Lei, Wang, & Guo, 2020), or real-time tennis ball tracking (Deepa, Tamilselvan, Abrar, & Sampath, 2019). In the competitions, both two architectures lead in some fields, without one

overperforming the other. The author will conduct experiments with SSD models for future research and compare the two architecture's accuracy and speed.

7.3. Concluding Remarks

This research's experiments achieved surprisingly good results (AP>90%) than the author expected (AP>80%). Though the research has proved that the YOLO technique is competent in bird sound detection tasks, it has limitations due to the lack of time. The author would like to explore the SSD models and compare the two architectures' accuracy and speed, which will start in July 2021.

References

Alexey. (2019). Darknet for Windows & Linex. Retrieved from https://github.com/AlexeyAB/darknet. Retrieved 2021, from Github https://github.com/AlexeyAB/darknet

Alexey. (2020). Yolov4_darknet. Retrieved from Github: https://github.com/kiyoshiiriemon/yolov4_darknet/tree/master/cfg

Almeida, L. B. (1994). The fractional Fourier transform and time-frequency representations. *IEEE Transactions on signal processing, 42*(11), 3084-3091.

Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., . . . Asari, V. K. (2018). The history began from alexnet: A comprehensive survey on deep learning approaches. *arXiv preprint arXiv:1803.01164*.

Aytar, Y., Vondrick, C., & Torralba, A. (2016). Soundnet: Learning sound representations from unlabeled video. *arXiv preprint arXiv:1610.09001*.

Baskerville, R., Baiyere, A., Gregor, S., Hevner, A., & Rossi, M. (2018). Design science research contributions: finding a balance between artifact and theory. *Journal of the Association for Information Systems, 19*(5), 3.

Bibby, C. J. (1999). Making the most of birds as environmental indicators. *Ostrich, 70*(1), 81-88.

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.

Brighten, A. (2015). *Vocalisations of the New Zealand morepork (Ninox novaeseelandiae) on Ponui Island : a thesis presented in partial fulfilment of the requirements for the degree of Master of Science in Zoology at Massey University, Palmerston North, New Zealand.* (Master of Science (M.Sc.) Masters). Massey University, Retrieved from http://hdl.handle.net/10179/7321

de Benito-Gorron, D., Lozano-Diez, A., Toledano, D. T., & Gonzalez-Rodriguez, J. (2019). Exploring convolutional, recurrent, and hybrid deep neural networks for speech and music detection in a large audio dataset. *EURASIP Journal on Audio, Speech, and Music Processing, 2019*(1), 1-18.

Deepa, R., Tamilselvan, E., Abrar, E., & Sampath, S. (2019). *Comparison of Yolo, SSD, Faster RCNN for real time tennis ball tracking for action decision networks.* Paper presented at the 2019 International Conference on Advances in Computing and Communication Engineering (ICACCE).

Domhan, T., Springenberg, J. T., & Hutter, F. (2015). *Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves.* Paper presented at the Twenty-Fourth International Joint Conference on Artificial Intelligence.

Fairbrass, A. J., Firman, M., Williams, C., Brostow, G. J., Titheridge, H., Jones, K. E., & Isaac, N. (2019). CityNet—Deep learning tools for urban ecoacoustic assessment. *Methods in Ecology & Evolution, 10*(2), 186-197. doi:10.1111/2041-210X.13114

Fang, Y., Ma, Z., Zhang, Z., Zhang, X.-Y., & Bai, X. (2017). *Dynamic multi-task learning with convolutional neural network.* Paper presented at the IJCAI.

Flanagan, J. L., & Golden, R. (1966). Phase vocoder. *Bell System Technical Journal, 45*(9), 1493-1509.

Florentin, J., Dutoit, T., & Verlinden, O. (2020). Detection and identification of European woodpeckers with deep convolutional neural networks. *Ecological Informatics, 55*. doi:10.1016/j.ecoinf.2019.101023

Forest & Bird. (2018, 29 May, 2018). How to identify New Zealand birds. Retrieved from https://www.forestandbird.org.nz/resources/how-identify-new-zealand-birds

Fu, S.-W., Tsao, Y., Lu, X., & Kawai, H. (2017). *Raw waveform-based speech enhancement by fully convolutional networks.* Paper presented at the 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC).

Fukushima, K. (1988). Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural networks, 1*(2), 119-130.

Girshick, R. (2015). *Fast r-cnn.* Paper presented at the Proceedings of the IEEE international conference on computer vision.

Goldkuhl, G. (2004). Design theories in information systems-a need for multi-grounding. *Journal of Information Technology Theory and Application (JITTA), 6*(2), 7.

Golik, P., Tüske, Z., Schlüter, R., & Ney, H. (2015). *Convolutional neural networks for acoustic modeling of raw time signal in LVCSR.* Paper presented at the Sixteenth annual conference of the international speech communication association.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Deep residual learning for image recognition.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

Heitzman, A. (2019). *Data visualization: What it is, why it's important & how to use it for SEO*. Retrieved from

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS quarterly*, 75-105.

Hong, S.-J., Han, Y., Kim, S.-Y., Lee, A.-Y., & Kim, G. (2019). Application of deep-learning methods to bird detection using unmanned aerial vehicle imagery. *Sensors, 19*(7), 1651.

Hunt, T. D., Nikora, M., & Blackbourn, C. (2019). *Analysis of morepork vocalizations recorded using a permanently located mobile phone*: CITRENZ.

Huzaifah, M. (2017). Comparison of time-frequency representations for environmental sound classification using convolutional neural networks. *arXiv preprint arXiv:1706.07156*.

Incze, A., Jancsó, H.-B., Szilágyi, Z., Farkas, A., & Sulyok, C. (2018). *Bird sound recognition using a convolutional neural network.* Paper presented at the 2018 IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY).

Indulska, M., & Recker, J. (2010). 13. Design science in IS research: a literature analysis. *Information systems foundations: The role of design science*, 285-302.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In.

Jaccard, P. (1912). The distribution of the flora in the alpine zone. 1. *New phytologist, 11*(2), 37-50.

Jiang, Z., Zhao, L., Li, S., & Jia, Y. (2020). Real-time object detection method based on improved YOLOv4-tiny. *arXiv preprint arXiv:2011.04244*.

Kahl, S., Clapp, M., Hopping, W., Goëau, H., Glotin, H., Planqué, R., . . . Joly, A. (2020). *Overview of birdclef 2020: Bird sound recognition in complex acoustic environments.* Paper presented at the CLEF 2020.

Kahl, S., Wilhelm-Stein, T., Klinck, H., Kowerko, D., & Eibl, M. (2018). Recognizing birds from sound - The 2018 BirdCLEF baseline system. In.

Kanjee, R. (2020). YOLOv5 controversy — Is YOLOv5 real? Retrieved from Start it up: https://medium.com/swlh/yolov5-controversy-is-yolov5-real-20e048bebb08

Koh, C.-Y., Chang, J.-Y., Tai, C.-L., Huang, D.-Y., Hsieh, H.-H., & Liu, Y.-W. (2019). *Bird sound classification using convolutional neural networks.* Paper presented at the CLEF (Working Notes).

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems, 25*, 1097-1105.

LeCun, Y., Haffner, P., Bottou, L., & Bengio, Y. (1999). Object recognition with gradient-based learning. In *Shape, contour and grouping in computer vision* (pp. 319-345): Springer.

Li, M., Zhang, Z., Lei, L., Wang, X., & Guo, X. (2020). Agricultural greenhouses detection in high-resolution satellite images based on convolutional neural networks: Comparison of Faster R-CNN, YOLO v3 and SSD. *Sensors, 20*(17), 4938.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). *Ssd: Single shot multibox detector.* Paper presented at the European conference on computer vision.

Loris, N., Yandre, M. G. C., Rafael, L. A., Rafael, B. M., Sheryl, B., & Carlos, N. S. (2020). Ensemble of convolutional neural networks to improve animal audio classification. *EURASIP Journal on Audio, Speech, and Music Processing, 2020*(1), 1-14. doi:10.1186/s13636-020-00175-3

Mohanty, R., Mallik, B. K., & Solanki, S. S. (2020). Automatic bird species recognition system using neural network based on spike. *Applied Acoustics, 161*. doi:10.1016/j.apacoust.2019.107177

Montavon, G., Orr, G., & Müller, K.-R. (2012). Neural networks-tricks of the trade second edition. *Springer, DOI, 10*, 978-973.

Montgomery, D. C. (2017). *Design and analysis of experiments*: John wiley & sons.

Moon, S., Song, H.-J., Sharma, V. D., Lyons, K. E., Pahwa, R., Akinwuntan, A. E., & Devos, H. (2020). Classification of Parkinson's disease and essential tremor based on gait and balance characteristics from wearable motion sensors: A data-driven approach. *medRxiv*, 2020.2004.2017.20065441. doi:10.1101/2020.04.17.20065441

Morera, Á., Sánchez, Á., Moreno, A. B., Sappa, Á. D., & Vélez, J. F. (2020). SSD vs. YOLO for detection of outdoor urban advertising panels under multiple variabilities. *Sensors, 20*(16), 4587.

Morgan, D. K. J., & Styche, A. (2012). Results of a community-based acoustic survey of ruru (moreporks) in Hamilton city. *Notornis, 29*, 123-129.

Murugan, P. (2017). Hyperparameters optimization in deep convolutional neural network/bayesian approach with gaussian process prior. *arXiv preprint arXiv:1712.07233*.

Nanni, L., Maguolo, G., & Paci, M. (2020). Data augmentation approaches for improving animal audio classification. *Ecological Informatics, 57*. doi:10.1016/j.ecoinf.2020.101084

Nanni, L., Rigo, A., Lumini, A., & Brahnam, S. (2020). Spectrogram classification using dissimilarity space. *Applied Sciences (2076-3417), 10*(12), 4176. Retrieved from http://wintec.idm.oclc.org/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edb&AN=144483952&site=eds-live&scope=site

New Zealand Birds Online. (2013). Morepork. Retrieved from New Zealand Birds Online: The digital encyclopaedia of New Zealand birds: http://nzbirdsonline.org.nz/species/morepork#bird-sounds

New Zealand Tourism. (2018). Bird conservation in New Zealand. Retrieved from https://media.newzealand.com/en/story-ideas/bird-conservation-in-new-zealand/

O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.

Oehlert, G. W. (2010). *A first course in design and analysis of experiments*.

Oikarinen, T., Srinivasan, K., Meisner, O., Hyman, J. B., Parmar, S., Fanucci-Kiss, A., . . . Feng, G. (2019). Deep convolutional network for animal sound classification and source attribution using dual audio recordings. *The journal of the acoustical society of america, 145*(2), 654-662.

Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., . . . Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*.

Oreilly. (2021). Evaluating detection (Intersection over union). Retrieved from https://www.oreilly.com/library/view/hands-on-convolutional-neural/9781789130331/a0267a8a-bd4a-452a-9e5a-8b276d7787a0.xhtml

Pacific Northwest Seismic Network. (N.A.). What is a Spectrogram?

.

Palaz, D., Collobert, R., & Doss, M. M. (2013). Estimating phoneme class conditional probabilities from raw speech signal using convolutional neural networks. *arXiv preprint arXiv:1304.1018*.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). *You only look once: Unified, real-time object detection.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

Redmon, J., & Farhadi, A. (2017). *YOLO9000: better, faster, stronger.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Ruff, Z. J., Lesmeister, D. B., Duchac, L. S., Padmaraju, B. K., Sullivan, C. M., Pettorelli, N., & Lecours, V. (2020). Automated identification of avian vocalizations with deep convolutional neural networks. *Remote Sensing in Ecology & Conservation, 6*(1), 79. Retrieved from http://wintec.idm.oclc.org/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edb&AN=142291148&site=eds-live&scope=site

Sainath, T. N., Mohamed, A.-r., Kingsbury, B., & Ramabhadran, B. (2013). *Deep convolutional neural networks for LVCSR.* Paper presented at the 2013 IEEE international conference on acoustics, speech and signal processing.

Sejdić, E., Djurović, I., & Jiang, J. (2009). Time–frequency feature representation using energy concentration: An overview of recent advances. *Digital signal processing, 19*(1), 153-183.

Sichkar, V. (2021). Training YOLO v3 for objects detection with custom data. Retrieved from Udemy: https://www.udemy.com/course/training-yolo-v3-for-objects-detection-with-custom-data/

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. In.

Sprengel, E., Jaggi, M., Kilcher, Y., & Hofmann, T. (2016). Audio based bird species identification using deep learning techniques. *No. CONF*, 547-559.

Stevens, S. S., Volkmann, J., & Newman, E. B. (1937). A scale for the measurement of the psychological magnitude pitch. *The journal of the acoustical society of america, 8*(3), 185-190.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., . . . Rabinovich, A. (2014). Going deeper with convolutions. In.

The Cacophony Project. (2021, 25 Febrary). Cacophony API. Retrieved from https://api.cacophony.org.nz

The Cacophony Project. (N.A.). The Cacophony Project. Retrieved from https://cacophony.org.nz/

Thyng, K. M., Greene, C. A., Hetland, R. D., Zimmerle, H. M., & DiMarco, S. F. (2016). True colors of oceanography: Guidelines for effective and accurate colormap selection. *Oceanography, 29*(3), 9-13.

Tubaro, P. L., & Mindlin, G. B. (2019). A dynamical system as the source of augmentation in a deep learning problem. *Chaos, Solitons & Fractals: X, 2*(-). doi:10.1016/j.csfx.2019.100012

Vaishnavi, V. K., & Kuechler, W. (2015). *Design science research methods and patterns: innovating information and communication technology*: Crc Press.

Winter, R. (2008). Design science research in Europe. *European Journal of Information Systems, 17*(5), 470-475.

WordPress. (2011). Tag: average precision. Retrieved from https://sanchom.wordpress.com/tag/average-precision/

Xie, J.-j., Ding, C.-q., Li, W.-b., & Cai, C.-h. (2018). Audio-only bird species automated identification method with limited training data based on multi-channel deep convolutional neural networks. *arXiv preprint arXiv:1803.01107*.

Xie, J., Hu, K., Zhu, M., Yu, J., & Zhu, Q. (2019). Investigation of different CNN-based models for improved bird sound classification. *IEEE Access, Access, IEEE, 7*, 175353-175361. doi:10.1109/ACCESS.2019.2957572

Xie, J., & Zhu, M. (2019). Handcrafted features and late fusion with deep learning for bird sound classification. *Ecological Informatics, 52*, 74-81. doi:10.1016/j.ecoinf.2019.05.007

Yates, F. (1978). *The design and analysis of factorial experiments*: Imperial Bureau of Soil Science Harpenden, UK.

Zhang, X., Yang, W., Tang, X., & Liu, J. (2018). A fast learning method for accurate and robust lane detection using two-stage feature extraction with YOLO v3. *Sensors, 18*(12), 4308.

Zhong, M., LeBien, J., Campos-Cerqueira, M., Dodhia, R., Lavista Ferres, J., Velev, J. P., & Aide, T. M. (2020). Multispecies bioacoustic classification using transfer learning of deep convolutional neural networks with pseudo-labeling. *Applied Acoustics, 166*. doi:10.1016/j.apacoust.2020.107375